

**PENERAPAN METODE *IMAGE STITCHING* PADA APLIKASI  
PEMBUATAN PETA GEOGRAFIS BERBASIS WEB DENGAN  
MENGUNAKAN ALGORITMA ORB (*ORIENTED FAST AND  
ROTATED BRIEF*)**

**TUGAS AKHIR**



**DAVID TRI SETYO WICAKSONO**

**NIM : 311810011**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI DAN DESAIN  
UNIVERSITAS MA CHUNG  
MALANG  
2024**

LEMBAR PENGESAHAN

TUGAS AKHIR

PENERAPAN METODE IMAGE STITCHING PADA APLIKASI  
PEMBUATAN PETA GEOGRAFIS BERBASIS WEB DENGAN  
MENGUNAKAN ALGORITMA ORB (ORIENTED FAST AND  
ROTATED BRIEF)

Oleh:

DAVID TRI SETYO WICAKSONO

311810011

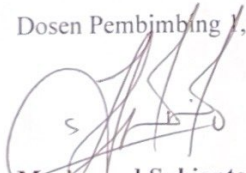
dari:

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS DESAIN DAN TEKNOLOGI UNIVERSITAS MA CHUNG

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan  
dan berhak mendapatkan gelar Sarjana Komputer (S.Kom)

Dosen Pembimbing 1,



Mochamad Subianto, S.Kom.,

M.Cs.

NIP. 20100002

Dosen Pembimbing 2,



Windra Swastika, S.Kom., MT.,

Ph.D.

NIP. 20100005

Dekan Fakultas Desain dan Teknologi,



Prof. Dr. Eng. Rany Budhi Widodo

NIP. 20070035

## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya nyatakan isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Penerapan Metode *Image Stitching* Pada Aplikasi Pembuatan Peta Geografis Berbasis Web Dengan Menggunakan Algoritma ORB (*Oriented FAST And Rotated BRIEF*) adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku

Malang, 20 Agustus 2024



A handwritten signature in black ink, appearing to read "David Tri Setyo Wicaksono".

David Tri Setyo Wicaksono  
NIM. 311810011

# **PENERAPAN METODE *IMAGE STITCHING* PADA APLIKASI PEMBUATAN PETA GEOGRAFIS BERBASIS WEB DENGAN MENGGUNAKAN ALGORITMA ORB (*ORIENTED FAST AND ROTATED BRIEF*)**

David Tri Setyo Wicaksono  
Universitas Ma Chung

## **Abstrak**

Pembuatan suatu peta geografis, terdapat berbagai macam cara, salah satunya adalah mengambil foto udara dari UAV(*Unmanned Aerial Vehicle*). Gambar UAV yang diambil dari berbagai sudut pandang dapat digabungkan menjadi peta yang lebih komprehensif dan akurat teknik *image stitching*. Metode *Image stitching* dilakukan dengan menggunakan algoritma ORB (*Oriented FAST and Rotated BRIEF*) dan diterapkan melalui aplikasi berbasis web. Perancangan aplikasi ini menerapkan metode *image stitching* berbasis web dan menjadi aplikasi alternatif untuk memudahkan pengguna dalam menggabungkan gambar. Pada tahapan pengujian, hasil uji fungsional aplikasi *image stitching* dan *ground truth* dengan algoritma ORB dapat menampilkan suatu peta geografis dan hasil uji *ground truth* dengan SSIM menggunakan 3 data gambar menghasilkan persentase yang kurang baik. Namun dalam kecepatan komputasi proses *stitching* menghasilkan waktu yang sangat cepat. Kesimpulannya bahwa algoritma ORB mendapatkan waktu komputasi yang cepat dalam proses *stitching* dan hasil uji SSIM disimpulkan bahwa kedua gambar secara struktur adalah kurang baik.

**Kata Kunci :** *Image stitching*, UAV drone, Algoritma ORB, Aplikasi Web

# **IMPLEMENTATION OF IMAGE STITCHING IN A WEB-BASED MAP MAKING USING ORB ALGORITHM (*ORIENTED FAST AND ROTATED BRIEF*)**

David Tri Setyo Wicaksono  
Universitas Ma Chung

## **Abstract**

There are various ways to create a geographic map, one of which is to take aerial photos from a UAV (Unmanned Aerial Vehicle). UAV images taken from various angles can be combined into a more comprehensive and accurate map using image stitching techniques. The Image stitching method is carried out using the ORB (Oriented FAST and Rotated BRIEF) algorithm and is implemented through a web-based application. The design of this application applies a web-based image stitching method and becomes an alternative application to make it easier for users to combine images. At the testing stage, the results of the functional test of the image stitching and ground truth applications with the ORB algorithm can display a geographic map and the results of the ground truth test with SSIM using 3 data produce a percentage of images that are less good. However, in terms of computational speed, the stitching process produces a very fast time. The conclusion is that the ORB algorithm gets fast computation time in the stitching process and the SSIM test results conclude that both images are structurally less good.

**Keywords:** Image stitching, UAV drone, ORB Algorithm, Web Application, SIFT Algorithm,

## KATA PENGANTAR

Puji Syukur kehadiran Tuhan Yang Maha Esa karena atas rahmat dan hidayah-Nya sehingga penelitian ini dapat berjalan dengan baik, Laporan Tugas Akhir ini disusun untuk memenuhi salah satu persyaratan kelulusan bagi Mahasiswa Universitas Ma Chung Malang. Atas dukungan moral dan materi yang telah diberikah dalam penyusunan laporan ini, penulis menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu terutama kepada:

1. Orang tua yang telah memberi dukungan moral serta doa dalam pelaksanaan tugas akhir ini.
2. Ibu Dr. Kestrilia Rega Prilianti, M.Si. Dekan Fakultas Sains dan Teknologi Universitas Ma Chung.
3. Bapak Mochammad Subianto, S.Kom., M.Cs. Dosen prodi Teknik Informatika selaku dosen pembimbing 1 penelitian Tugas Akhir.
4. Bapak Windra Swastika, S.Kom., MT., Ph.D.. Dosen prodi Teknik Informatika selaku dosen pembimbing 2 penelitian Tugas Akhir.
5. Yang saya kasihi, Adenia Novenda Putri Pramadewi, yang sudah banyak kontribusi semangat dan motivasi untuk tetap maju dan tidak gentar selama mengerjakan Tugas Akhir
6. Saudara dan teman-teman yang telah memberikan dukungan moral. Terutama teman baru saya Shashank Gupta yang membantu saya dalam menyelesaikan error dan debugging

Akhir kata, penulis berharap Tuhan YME membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan Praktik Kerja Lapangan ini dapat bermanfaat dan menambah ilmu pembaca.

Malang 7 Juli 2024

David Tri Setyo Wicaksono

## DAFTAR ISI

ABSTRAK .....	I
<i>ABSTRACT</i> .....	II
KATA PENGANTAR .....	III
DAFTAR ISI .....	IV
DAFTAR GAMBAR .....	VII
DAFTAR TABEL .....	IX
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Identifikasi Masalah .....	3
1.3. Batasan Masalah .....	3
1.4. Perumusan Masalah .....	3
1.5. Tujuan Penelitian .....	3
1.6. Luaran .....	4
1.7. Manfaat .....	4
1.8. Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA .....	6
2.1. Peta .....	6
2.1.1 Definisi Peta .....	6
2.1.2 Fungsi Peta .....	7
2.1.3 Peta digital .....	7
2.1.4 Jenis - Jenis Peta .....	8
2.2. <i>Unmanned Aerial Variable</i> .....	9
2.2.1 Pengenalan <i>UAV/Drone</i> .....	10
2.2.2 Teknik Fotogrametri .....	11
2.3. <i>Python</i> .....	12

2.3.1. <i>Framework Django</i> .....	12
2.3.2. <i>OpenCV</i> .....	13
2.4. Algoritma ORB .....	14
2.4.1. FAST( <i>Features from Accelerated Segment Test</i> ) Detector .....	15
2.4.2 <i>Orientation by Intensity Centroid</i> .....	16
2.4.3 BRIEF .....	16
2.5. Algoritma SIFT .....	17
2.6. <i>Image stitching</i> .....	18
2.6.1 Proses Image stitching .....	18
2.7. Web .....	23
BAB III PERANCANGAN APLIKASI .....	25
3.1. Alur Penelitian.....	25
3.2. Identifikasi Masalah .....	27
3.3. Analisis Kebutuhan .....	27
3.3.1 Kebutuhan Pengguna .....	27
3.3.2 Kebutuhan Peneliti.....	27
3.4. Studi Pustaka .....	28
3.5. Pengumpulan Data .....	30
3.6. Desain Sistem .....	31
3.6.1 Alur Aplikasi .....	31
3.6.2 <i>Mockup</i> Aplikasi dan Pembuatan Aplikasi .....	32
3.7. Rancangan Pengujian Sistem .....	34
3.7.1. Rancangan pengujian Fungsionalitas <i>Image stitching</i> .....	35
3.7.2. Rancangan pengujian <i>Ground Truth</i> .....	35
BAB IV HASIL DAN PEMBAHASAN .....	37
4.1. Sistem <i>Image Stitching</i> .....	37



4.2. Perancangan Aplikasi .....	39
4.2.1 Model <i>Forms</i> .....	40
4.2.2 <i>Image Pre-processing</i> .....	41
4.2.3 <i>Feature Extraction</i> .....	42
4.2.4 <i>Homography Estimation (RANSAC)</i> .....	45
4.3. Hasil Pengujian.....	45
4.3.1 Hasil Pengujian Fungsionalitas Image Stitching .....	46
4.3.3 Hasil Pengujian <i>Ground Truth</i> .....	49
BAB V KESIMPULAN DAN SARAN.....	55
5.1 Kesimpulan.....	55
5.2 Saran .....	55
DAFTAR PUSTAKA .....	56

## **DAFTAR TABEL**

Tabel 3. 1 Hasil Ringkasan dari Penelitian Terdahulu .....	28
Tabel 3. 2 Daftar Data gambar UAV/Drone .....	30
Tabel 4. 1 Hasil Analisis Tiga Objek Gambar Berbeda.....	53

## DAFTAR GAMBAR

Gambar 2. 1 Proses mapping dengan drone .....	10
Gambar 2. 2 Perencanaan menggunakan teknik fotogrametri .....	11
Gambar 2.3 Interest Point Algoritma ORB .....	15
Gambar 2.4 Image Gradient (kiri), Keypoint Descriptor (kanan) .....	17
Gambar 2.5 Image Gradient (kiri), Keypoint Descriptor (kanan) .....	17
Gambar 2.6 Sistem Koordinat Kamera .....	19
Gambar 2.7 Sudut Pandang Horizontal dan Vertikal .....	19
Gambar 2.8 Akuisisi dengan Rotasi Kamera .....	20
Gambar 2.9 <i>Overlap</i> Dua Gambar .....	21
Gambar 2.10 Konversi Gambar ke Bentuk Bola .....	21
Gambar 3. 1 Alur Penelitian .....	25
Gambar 3. 2 <i>Activity Diagram User</i> .....	32
Gambar 3. 3 <i>Landing Page</i> .....	33
Gambar 3. 4 Halaman file sudah terupload .....	33
Gambar 3. 5 Hasil <i>Stitching</i> .....	34
Gambar 4. 1 Sistem <i>Image Stitching</i> dengan Algoritma ORB .....	37
Gambar 4. 2 Dataset UAV original (sisi kanan) diperkecil (sisi kiri) .....	40
Gambar 4.3 Gambar yang Digayscale .....	42
Gambar 4.4 Deteksi <i>keypoint</i> .....	43
Gambar 4.5 Dekripsi <i>keypoint</i> .....	43
Gambar 4.6 Hasil Pencarian kecocokan <i>Keypoint</i> Gambar 1 dan 2 .....	44
Gambar 4.7 Hasil Mencocokkan <i>Keypoint</i> dengan RANSAC .....	45
Gambar 4.8 Landing page dari aplikasi .....	46
Gambar 4.9 User upload gambar yang sudah diperkecil .....	47
Gambar 4.10a Hasil Unggahan pertama .....	47
Gambar 4.10b Hasil Unggahan Kedua .....	47
Gambar 4.10c Hasil Unggahan Ketiga .....	47
Gambar 4.11 Hasil <i>Stitching</i> Proyek Tol Yogyakarta .....	48
Gambar 4.12 Hasil <i>Stitching</i> Data Gambar Hutan .....	48
Gambar 4.13 Hasil <i>Stitching</i> Data Gambar Pabrik .....	48
Gambar 4.14 Dasar Pengujian <i>Ground Truth</i> .....	49

Gambar 4.15	Bagian atas adalah map asli dan Bagian bawah adalah map hasil proses <i>stitching</i> .....	50
Gambar 4.16	Pencocokan Gambar Asli dan Gambar Hasil <i>Stitching</i> Proyek Tol .....	51
Gambar 4.17	Pencocokan Gambar Asli dengan Gambar Hasil <i>Stitching</i> Mission 1 .....	51
Gambar 4.18	Pencocokan Gambar Asli dengan Gambar Hasil <i>Stitching</i> Quarry .....	51
Gambar 4.19	Hasil SSIM projektol Yogyakarta sebesar 33% .....	52
Gambar 4.20	Hasil SSIM mission1 Yogyakarta sebesar 34% .....	52
Gambar 4.21	Hasil Uji SSIM Data Quarry Dari Kaggle Sebesar 35% .....	53

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Suatu sistem untuk mengolah data seperti proses, mendapatkan data, menyusun, simpan, manipulasi data dengan berbagai cara untuk menghasilkan informasi yang terbaik yaitu informasi yang relevan, akurat dan *ontime* yang digunakan untuk keperluan individu, bisnis maupun pemerintahan untuk mengambil keputusan disebut dengan teknologi (Cholik, 2021). Perkembangan teknologi menimbulkan dampak yang besar terhadap pembuatan suatu peta geografis. Pembuatan peta geografis memerlukan penggabungan gambar yang akurat dan terperinci. Salah satu tantangannya adalah penggabungan gambar-gambar menjadi satu tampilan panoramik yang konsisten.

Rangkaian terpadu dari empat kelompok informasi yakni titik, garis, wilayah dan nama yang membentuk ciri, pola, bentuk, ukuran, ketebalan simbol disebut peta (Mango, 2019). Dalam pembuatan suatu peta geografis, terdapat berbagai macam cara, salah satunya adalah mengambil foto udara dari UAV (*Unmanned Aerial Vehicle*). Teknik *image stitching* menggabungkan gambar UAV yang diambil dari berbagai sudut pandang dapat digabungkan menjadi peta yang lebih komprehensif dan akurat. Menurut Oktaviano *et al* (2021) proses mengombinasikan gambar berserta bidangnya yang bertumpukan, pada gambar pertama yang mempunyai kesamaan gambar kedua untuk memperoleh hasil yang beresolusi tinggi. Salah satu teknik yang sering dipakai adalah *image stitching* menggunakan algoritma ORB (*Oriented FAST and Rotated BRIEF*).

Algoritma ORB (*Oriented FAST and Rotated BRIEF*) yaitu kombinasi antara *Oriented FAST* dengan *Rotated BRIEF*. Kedua algoritma tersebut memiliki kelebihan dalam kecepatan mengkalkulasi dimana lebih cepat daripada algoritma SIFT (*Scale-Invariant Feature Transform*) dan SURF (*Speeded-Up Robust*

*Features*) menurut Zhang hua *et al.*, (2020). *Image stitching* dalam mengekstrak *feature points* suatu gambar untuk meningkatkan akurasi kecocokan, maka dibantu oleh suatu algoritma RANSAC (*Random Sample Consensus*) untuk memastikan pemrosesan *real-time* yang kuat dan ekstraksi *feature point* yang cepat.

Metode tradisional dalam pembuatan peta geografis secara *real-time* dapat dilakukan dengan menggunakan satelit dari ketinggian yang tinggi (Goh J.N *et al*, 2021) . Metode ini memiliki kekurangan tidak mudah digunakan oleh masyarakat untuk membuat peta geografis di daerah tertentu secara *real-time*. Sehingga untuk memudahkan masyarakat dalam membuat peta geografis, dapat digunakan UAV (*Unmanned Aerial Vehicle*) untuk mengambil beberapa gambar dengan ketinggian 50 - 140 meter dari permukaan, tergantung kamera dari UAV itu sendiri. Selanjutnya, proses *image stitching* pembuatan peta dapat dilakukan melalui aplikasi berbasis web sekaligus menggunakan algoritma ORB.

Hampir semua organisasi maupun institusi memiliki jaringan internet untuk menyediakan informasi kepada masyarakat (Widyaningtyas *et al*, 2018). *Image stitching* akan diterapkan ke dalam suatu aplikasi berbasis web untuk memudahkan masyarakat dalam menggabungkan gambar menjadi peta secara *online*. Selain pembuatan peta secara *online*, terdapat aplikasi yang serupa. Namun kekurangannya adalah tidak semua kalangan masyarakat paham akan pemakaian aplikasi tersebut. Seperti pada contohnya aplikasi dekstop ArcGIS, ptGUI, dsb. Hal serupa juga sulit mencari aplikasi *image stitching online* di *browser*. Maka dari itu, tujuan dari pengembangan web ini yaitu memudahkan masyarakat untuk mengakses secara gratis dan dapat menggabungkan beberapa gambar menjadi satu gambar panorama atau map.

Oleh karena itu, penelitian ini menerapkan metode *image stitching* pada aplikasi berbasis web dengan menggunakan algoritma ORB pada pembuatan peta geografis dengan gambar UAV. Dengan demikian, akan dihasilkan suatu peta yang *real-time* dengan cakupan yang luas dan lebih kaya akan informasi geografis. Penerapan teknologi ini diharapkan dapat mendukung berbagai aplikasi dalam pemetaan, pemantauan lingkungan, manajemen sumber daya, serta penelitian ilmiah yang memerlukan data spasial dengan presisi akurat.

## 1.2. Identifikasi Masalah

Berdasarkan latar belakang terkait penerapan *image stitching* pada aplikasi berbasis web, peneliti menemukan suatu masalah yakni terbatasnya akses *real-time* ke data geografis dan kurangnya visualisasi data geografis. Selain kedua hal tersebut, ditemukan juga masalah seperti bagaimana cara membuat suatu aplikasi berbasis web yang dapat menerapkan metode *image stitching*.

## 1.3. Batasan Masalah

Tugas akhir ini memiliki batasan masalah sebagai berikut.

1. Data input gambar diambil menggunakan drone dan terdapat di *website* yang menyediakan data UAV geografis
2. Gambar diunggah yang digunakan adalah gambar *drone* dengan format RGB.
3. Proses penggabungan gambar *drone* dilakukan menggunakan metode ORB.
4. Orientasi penggabungan gambar hanya bisa arah horizontal atau vertikal.

## 1.4. Perumusan Masalah

Berdasarkan identifikasi masalah, didapatkan suatu rumusan masalah yakni, bagaimana cara memproses dan mengimplementasikan suatu algoritma ORB (*Oriented FAST and Rotated BRIEF*) ke aplikasi web yang *real-time* dan dapat memvisualisasikan hasil peta geografis dengan menerapkan metode *image stitching*?

## 1.5. Tujuan Penelitian

Pengerjaan tugas akhir ini memiliki tujuan yaitu :

1. Merancang sebuah aplikasi berbasis web dengan menerapkan *image stitching* berbasis web yang *real-time* dan dapat memvisualkan hasil peta geografis dari gambar drone dengan metode ORB.
2. Menjadi aplikasi alternatif untuk memudahkan pengguna dalam menggabungkan gambar dari *drone*.

## **1.6. Luaran**

Halaman website dengan sistem informasi geografis yang menerima suatu data gambar *drone* yang diolah menggunakan algoritma ORB (*Oriented FAST and Rotated BRIEF*) untuk melakukan penggabungan data gambar menjadi suatu peta dan hasil publikasi dalam bentuk jurnal adalah luaran dari tugas akhir ini.

## **1.7. Manfaat**

Tugas akhir ini memiliki manfaat yaitu:

1. Bagi Universitas Ma Chung, khususnya bagi program studi Teknik Informatika yaitu dapat menghasilkan lulusan yang siap dalam pekerjaan dan bermanfaat bagi lingkungan masyarakat sesuai dengan 12 visi dan misi Universitas Ma Chung.
2. Bagi peneliti selanjutnya, dapat melakukan pengembangan dan penyempurnaan untuk mencapai hasil untuk mencapai hasil yang baik.
3. Bagi penulis, dapat menambah wawasan dalam pengelolaan dalam penggabungan beberapa data gambar drone dengan menggunakan metode ORB.

## **1.8. Sistematika Penulisan**

Sistematika dalam penulisan laporan penelitian ini dibagi kedalam 5 bab sebagai berikut.

1. Bab 1 Pendahuluan : penjelasan latar belakang dalam melakukan tugas akhir, yakni identifikasi masalah yang ada, batasan masalah yang akan dibahas, rumusan masalah, tujuan dari penelitian, manfaat penelitian ini dibuat, luaran/hasil akhir dari penelitian, sistematika penulisan tugas akhir, dan rencana penelitian yang akan dilaksanakan selama 4 bulan ke depan



2. Bab 2 Tinjauan Pustaka : menjelaskan dasar teori yang diperoleh dari buku, jurnal dari *website* yang berkaitan dengan penelitian ini.
3. Bab 3 Perancangan Aplikasi : menjelaskan bagaimana rancangan awal dari aplikasi berbasis web dalam menggabungkan data gambar (*image stitching*) drone dengan menggunakan algoritma ORB.
4. Bab 4 Hasil dan Pembahasan : menjelaskan *output* dari pembuatan sistem dan pengujian sistem. Pembuatan sistem dimulai sistem dari image stitching dengan algoritma ORB dan perancangan aplikasi. Tahap pengujian yang pertama dilakukan yaitu mengetahui fungsionalitas dari aplikasi *image stitching* berbasis web, kedua pengujian perbandingan kedua algoritma (ORB dan SIFT) dalam mengekstrak *feature points*, persentase kecocokan dan waktu komputasi dihitung dengan berapa detik.
5. Bab 5 Kesimpulan dan Saran : berisi ringkasan dan saran dari tugas akhir yang sudah dikerjakan. Saran ditujukan bagi peneliti berikutnya untuk mengembangkan hasil penelitian ini ke jenjang berikutnya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Peta**

##### **2.1.1 Definisi Peta**

Representasi visual dari potongan atau keseluruhan permukaan bumi pada bidang datar memakai ukuran dan proyeksi tertentu disebut peta. Penyajian elemen-elemen permukaan bumi pada peta dilakukan dengan memilih dan menggeneralisasi data sesuai dengan tujuan pembuatan peta. Peta menyajikan berbagai informasi mengenai permukaan bumi yang diharapkan dapat dimanfaatkan dengan baik oleh pengguna. Peta memiliki beberapa fungsi yaitu.

1. Menunjukkan suatu wilayah.
2. Menampilkan elemen yang ada di permukaan bumi.
3. Menunjukkan ukuran seperti jarak dan arah.
4. Mengumpulkan dan mengelompokkan data dari permukaan bumi.

Persyaratan- persyaratan geometrik yang harus dipenuhi oleh peta yang ideal adalah.

1. Mengukur titik dari peta harus selaras dengan titik sebenarnya.
2. Besarnya elemen yang digambarkan pada peta harus selaras dengan sebenarnya.
3. Arah garis yang digambarkan di peta harus selaras dengan arah sebenarnya.
4. Struktur yang direpresentasikan di peta harus sesuai dengan struktur yang sebenarnya.

Namun, mustahil merancang peta seperti yang dijelaskan karena lapisan bumi merupakan bidang yang tidak terstruktur. Jika salah satu syarat di atas terpenuhi, maka harus disesuaikan dengan tujuan pembuatan peta tersebut. Pada awalnya data geografi yang dilambangkan dengan titik, garis, dan poligon digambarkan pada sebuah kertas, sekarang data geografi tersebut dapat digambarkan melalui komputer. Peta yang dapat digambarkan pada komputer disebut peta digital. Walaupun masih terdapat kesamaan antara peta digital dengan

peta konvensional, ada beberapa yang membuat peta digital lebih unggul. Untuk memuat informasi dalam skala kecil, peta konvensional mudah dibaca, tetapi jika informasi yang ditampilkan dalam skala besar, maka pengguna akan kesulitan dalam menterjemahkan informasi yang disampaikan. Oleh karena itu, peta digital akan sangat berguna jika informasi yang ingin disampaikan dalam skala besar.

### **2.1.2 Fungsi Peta**

Dalam hal ini, peta memiliki fungsi diantaranya:

1. Menampilkan wilayah.
2. Menunjukkan ukuran seperti jarak, luas atau arah di atas lapisan bumi.
3. Menampilkan struktur seperti bentuk benua, negara, gunung, sungai, bangunan dan sebagainya.
4. Memperlihatkan gerakan dan kecenderungan perubahan.
5. Membantu peneliti dalam melakukan pemeriksaan kondisi di daerah tertentu.
6. Memberikan informasi tentang karakter ruang suatu daerah.

### **2.1.3 Peta digital**

Peta digital adalah suatu peta dari kertas, gambar satelit atau gambar - gambar yang terdigitalisasi yang nantinya akan digabungkan dengan gambar yang lain sehingga setelah diolah akan memberikan informasi. Data peta digital dibagi menjadi tiga bagian, yakni.

#### **1. Data Spasial**

Data spasial memiliki titik, garis, dan poligon. Titik mewakili segala sesuatu yang dapat digambarkan dengan koordinat x, y seperti lokasi dari obyek atau tempat. Garis mewakili segala sesuatu yang memiliki panjang seperti jalan. Sementara itu, poligon mewakili segala sesuatu yang memiliki batasan seperti kota, negara, dan lain-lain.

#### **2. Data Tabular**

Data tabular adalah data yang berisikan data spasial, contohnya data nama kecamatan pada peta wilayah

### 3. Data Gambar

Data bentuk gambar biasa digunakan untuk membuat peta berbentuk digital. Salah satu fungsinya sebagai peta dasar dan data tersebut bisa didapatkan dari hasil foto satelit, atau peta konvensional sudah dipindai.

#### 2.1.4 Jenis - Jenis Peta

Peta terdapat berbagai macam jenis, diantara lainnya adalah.

##### 1. Berdasarkan isinya

###### a. Peta umum

Peta ini yang menggambarkan kenampakan pada suatu daerah secara umum, baik kenampakan alami maupun budaya. Pada dasarnya, peta umum meliputi.

###### i. Peta topografi

Peta menggambarkan relief permukaan bumi dan secara umum digunakan garis kontur.

###### ii. Peta chorografi

Peta yang menggambarkan kenampakan daerah yang cukup luas, dan berskala kecil sampai sedang.

###### iii. Peta dunia

Peta berskala kecil berisikan hal - hal yang penting dan global.

###### b. Peta khusus

Peta ini biasa dipanggil peta tematik yang menggambarkan kenampakan - kenampakan tertentu pada permukaan bumi, baik kondisi fisik maupun budaya. Peta tematik diantara lainnya memiliki contoh seperti peta iklim, peta persebaran penduduk, peta pariwisata, dsb.

##### 2. Berdasarkan objeknya

Berdasarkan keadaan objek peta dapat dibedakan menjadi dua, peta dinamis dan statis.

###### a. Peta dinamis

Peta yang menggambarkan keadaan objek yang sifatnya labil atau mudah berubah. Sebagai contoh perkembangan permukaan, peta daerah tepian transmigrasi.

b. Peta statis

Peta yang menggambarkan keadaan objek bersifat relatif tetap. Sebagai contoh peta tanah, peta geologi.

## 2.2. Unmanned Aerial Variable

*Unmanned Aerial Vehicle* (UAV), adalah mesin terbang nirawak yang dikontrol dan terbang dengan sendirinya berdasarkan target wilayah yang sudah dimasukkan ke dalam program. UAV ini bisa dimodifikasi seperti memuat senjata maupun muatan lainnya.

Pemanfaatan UAV dibidang pemetaan sangat menjanjikan, antara lain: pemetaan daerah irigasi, pemetaan di daerah perkotaan, pemetaan untuk penyajian model 3 dimensi yang memungkinkan pengguna untuk melakukan perhitungan volume material atau pemetaan berskala besar lainnya.

UAV pada umumnya mampu terbang berdiameter tiga ribu meter untuk melakukan satu misi. Fitur dari pesawat ini adalah mampu membawa sensor salah satunya teknologi INS (*Inertial Navigation System*) / IMU(*Inertial Measurement Unit*) yang menghasilkan *orthophoto* dan DEM (*Digital Elevation Model*) dengan akurasi satuan centimeter (cm). Kualitas adalah hal yang krusial, namun kebutuhan harus disesuaikan dan berpeluang untuk menaikkan kualitas secara bertahap (Sudarman 2019). Masa pengolahan teknologi ini biasanya memperoleh lebih dari 500 Ha/hari sehingga menghadirkan potensi yang menguntungkan (Rokhmana, 2015). Proses dapat dimonitor secara langsung dengan *software* pendukung . Dalam mencermati uraian yang sudah disampaikan, dapat diambil 3 hal.

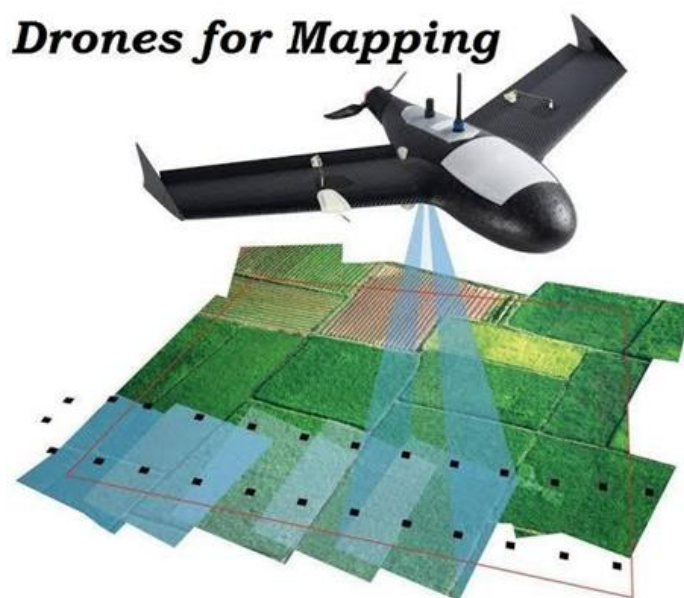
1. Spesifikasi teknis dan keamanan UAV,
2. Keterampilan pilot

3. Area yang disurvei, dengan menerapkan konsep FFP-LA (*Fit for Purpose Land Administration*) dalam arean <10.000 Ha. Hasil ekstraksi berbasis RGB (Red Green Blue)

### 2.2.1 Pengenalan UAV/Drone

Unmanned Aerial Vehicle (UAV), adalah sebuah mesin terbang yang dikontrol melalui *remote control* dengan jarak jauh atau pesawat terbang dengan sendirinya berdasarkan titik wilayah yang dimasukkan ke dalam program. UAV memiliki banyak manfaat seperti mengangkut barang, menyiram suatu lahan pertanian, mengambil foto terutama foto pemetaan daerah perkotaan sampai ruang lingkup perdesaan.

Pemanfaatan UAV dibidang pemetaan sangat menjanjikan, antara lain: pemetaan daerah irigasi, pemetaan di daerah perkotaan, pemetaan untuk penyajian model 3 dimensi yang memungkinkan pengguna untuk melakukan perhitungan volume material atau pemetaan berskala besar lainnya. Berikut konsep proses *mapping* dengan *drone*



Sumber <https://www.agrivi.com/wp-content/uploads/2018/03/Drone-Images.jpg>

Gambar 2. 1 Proses mapping dengan drone

### 2.2.2 Teknik Fotogrametri

Menurut Komang Sri H (2019) menyatakan bahwa fotogrametri merupakan ilmu, seni dan teknik untuk memperoleh data dan informasi tentang suatu objek atau fenomena melalui proses pencatatan, pengukuran dan interpretasi foto udara sedangkan pemetaan fotogrametri merupakan proses pemetaan objek - obiek di permukaan dengan menggunakan foto udara yang bertampalan (*overlap*) sebagai media, dimana proses interpretasi dan pengolahan geometri dilakukan untuk menghasilkan pea orthofoto dan peta garis. Dalam pembuatan peta 2 dan peta garis dengan geometri dan skala yang benar maka dilakukan proses restitusi foto udara secara tunggal (rektifikasi) maupun secara stereo (*orthophoto*), dimana proses ini memerlukan titik-titik referensi yang diketahui koordinatnya. Berikut gambar 2.2 adalah gambar untuk menentukan wilayah yang akan diambil.



Sumber [www.researchgate.com/figure/e-Waypoints-planning-with-the-ArduPilot-Mission-Planner\\_fig1\\_357151288](http://www.researchgate.com/figure/e-Waypoints-planning-with-the-ArduPilot-Mission-Planner_fig1_357151288)

Gambar 2. 2 Perencanaan menggunakan teknik fotogrametri

*Orthophoto* merupakan sebuah gambar yang telah dikoreksi secara geometrik agar sesuai yang ada di peta dan dengan representasi visual yang nyata. Menurut Kasser dan Polidori (2002) menyatakan bahwa ortofotografi adalah produk yang mampu mengambil alih kartografi kuno secara sempurna.

## 2.3. Python

Desember 1989 Oleh Guido Van, python adalah secara umum salah satu bahasa pemrograman yang memiliki level tinggi dan populer. Python adalah bahasa yang diketik secara dinamis dan biasa disebut sebagai “*duck typing*” di *software engineer*. Salah satu alasan bahasa ini terkenal, yakni keterbacaan terhadap kode. Sebagai programmer yang baik, penting seseorang dapat memahami kode yang sudah dibuat oleh programmer lain. Berikut adalah fitur programming menggunakan python.

Risyda F *et al.*, (2020) menyatakan bahwa bahasa permograman ini telah menjadi bahasa yang secara umum sering dibuat untuk aplikasi data sains. Bahasa python adalah gabungan dari *general purpose programming languages* dengan memudahkan kegunaan bahasa *domain specific scripting*. Python memiliki pustaka untuk pemuatan data, visualisasi, statistik, dan pemrosesan gambar,dll . Berikut adalah pustaka yang dipakai selama pengerjaan tugas akhir ini.

### 2.3.1. Framework Django

*Framework* Django sebuah *framework full-stack* untuk pembuatan aplikasi web dengan bahasa pemrograman Python. Menurut (Risyda F *et al.*, 2020) mengemukakan bahwa Django adalah kerangka kerja Python web tingkat tinggi yang mendorong pengembangan cepat dan desain pragmatis yang bersih. *Full-stack* artinya *framework* tersebut meliputi sisi *front-end* dan *back-end*. *Front-end* adalah bagian yang dari sisi pengguna. Sebaliknya, *back-end* yaitu bagian yang berhubungan dengan database dan logika.

Menurut Risyda F, *et al* (2020) Django adalah kerangka kerja Python Web tingkatan atas yang menstimulasi pengembangan cepat dan rancangan berorientasi hasil yang bersih. Kerangka kerja web adalah komponen yang menghadirkan prosedur untuk membangun situs dengan tanggap dan praktis. Tujuan utama Django adalah untuk meringankan pembuatan situs yang dipicu oleh basis data yang rumit. Beberapa situs terkenal yang memakai Django adalah Youtube, Quora, The Onion, Pinterest, dsb.



Adapun kelebihan dari framework Django sebagai berikut.

1. Merepresentasikan model data dari basis data yang memuat bentuk objek, sesuai dengan konsep pemrograman yang berbasis *object - oriented programming* (OOP).
2. Django mengerti dalam melindungi data sensitif, Django menyediakan tampilan sisi admin secara otomatis. Tampilan admin ini tidak dapat diakses oleh sembarang orang.
3. Penyusunan URL menjadi mudah dan fleksibel.
4. Menyediakan *template* yang kuat dan mudah dipelajari untuk pengembang web dan desainer web, serta mendukung *template hierarchy* (pewarisan).
5. Sistem *cache* mudah digunakan.
6. Mempermudah pengembang web dalam membuat situs multilingual.
7. Tidak memerlukan instalasi web server selama proses development

### 2.3.2. OpenCV

Menurut Muhammad & Nur (2022) menyatakan bahwa OpenCV adalah pustaka *Application Programming Interface* (API) yang terlatih dalam proses gambar dengan *Computer Vision*. Secara teori OpenCV digunakan seperti mencontoh kinerja visual manusia yakni dengan memandang objek lewat indera penglihatan dan gambar pada objek tersebut lalu dilanjutkan ke otak untuk mencerna sehingga paham objek apa yang nampak pada visual mata manusia. OpenCV adalah perpustakaan *Computer Vision open source* untuk C/C, OpenCV dirancang untuk aplikasi waktu nyata, memiliki fitur akuisisi gambar/video yang bagus (Surahman et al., 2021).

OpenCV mempunyai macam – macam fitur yang bermanfaat, berikut adalah macam – macam fitur dari OpenCV:

#### 1. *Image dan Video I/O*

Fitur yang mampu untuk memahami data gambar dari *file*, atau dari video langsung, serta pembuatan file gambar dan video.

#### 2. *Computer Vision dan Image Processing* (API tingkat *low* dan *mid* )

Antarmuka yang memungkinkan eksperimen berbagai algoritma *computer vision*, seperti deteksi garis, deteksi tepi garis, proyeksi elips, piramida gambar, pencocokan *template*, berbagai transformasi (Fourier, cosine diskrit, distance transform) dan lain sebagainya..

3. *Computer Vision High Level*

OpenCV juga dapat melakukan teknik tingkat tinggi, seperti mendeteksi objek, deteksi sidik jari, *optical flow*, dsb.

4. AI (*Artificial Intelligent*) dan ML(*Machine Learning*)

OpenCV mempunyai pustaka ML (*Machine Learning*), yang memfasilitasi komputer dalam mengidentifikasi, menetapkan keputusan dan melaksanakan aksi terhadap objek tertentu.

5. Sampling Gambar dan Substraksi

Fitur ini mencakup substraksi subregion dari gambar, *random sampling*, rotasi, dan lain sebagainya.

6. Gambar Biner

OpenCV memiliki fitur untuk dapat merancang dan mempelajari gambar biner.

7. Pemodelan 3D

Fitur ini sangat berguna untuk pemetaan dan lokalisasi, baik menggunakan stereo kamera maupun kamera tunggal dengan sudut pandang yang berbeda.

## 2.4. Algoritma ORB

ORB adalah algoritma dalam pustaka OpenCV yang digunakan untuk mengidentifikasi dan mendeskripsikan fitur. Algoritma ini dibuat gabungan algoritma FAST untuk mendeteksi fitur dan algoritma BRIEF untuk mendeskripsinya. Algoritma bisa menjadi alternatif yang kuat dan cepat dibanding memakai SIFT dan SURF.

Algoritma ORB berbasis pada nilai biner, dengan dua komponen utama yaitu FAST sebagai identifikasi *keypoint* dan konsep rotasi BRIEF yang disempurnakan untuk meningkatkan kemampuan dalam deteksi maupun mendeskripsi fitur. Berikut gambar 2.3 adalah hasil dari ekstraksi *interest point* atau *feature points* oleh ORB



Gambar 2.3 Interest Point Algoritma ORB

#### **2.4.1. FAST(*Features from Accelerated Segment Test*) Detector**

Detektor FAST sering digunakan dalam implementasi *computer vision*. Algoritma ini bertujuan untuk mendeteksi fitur dengan cara mendeteksi tepi dimana keunggulannya dalam komputasi yang efisien. Sesuai namanya, FAST *detector* adalah salah satu metode deteksi fitur yang cepat dibandingkan metode lain, sangat cocok digunakan untuk mendeteksi gambar berbasis *real-time* yaitu pada pengambilan gambar video. Dalam ORB, FAST menggunakan satu parameter yaitu intensitas *threshold* diantara piksel pusat dan sekelilingnya. Pendekatan FAST-9 yang memiliki kinerja baik yang digunakan dalam ORB. FAST tidak menghitung sudut maka digunakanlah perhitungan *Harris Corner* untuk menentukan *keypoint* FAST. Untuk mendapatkan sejumlah  $N$  *keypoint*, pertama diatur *threshold* seminim mungkin untuk mendapatkan *keypoint* yang lebih besar dari  $N$ , lalu diurut sesuai dengan *Harris Corner* dan mengambil titik  $N$  teratas. FAST tidak menghasilkan fitur dengan multiskala, sehingga skala piramid dibangun pada gambar dan FAST dihitung pada setiap *level* piramid.

### 2.4.2 Orientation by Intensity Centroid

Pendekatan yang digunakan untuk menghitung orientasi ialah dengan *intensity centroid*. Asumsinya ialah intensitas sudut seimbang dari titik pusat dan *vector* ini digunakan untuk menghitung orientasi.

Definisi moments:

$$m_{pq} = \sum_{x,y} x^p x^q I(x, y) \quad (1)$$

Dengan moments tersebut mungkin ditemukan centroid:

$$C = \left( \frac{m_{10} m_{01}}{m_{00} m_{00}} \right) \quad (2)$$

Lalu membangun *vector* dari sudut pusat O ke centroid, OC. Orientasi *patch* sebagai berikut:

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (3)$$

Dimana atan2 merupakan quadran arctan.

### 2.4.3 BRIEF

BRIEF merupakan sebuah metode yang digunakan untuk mendeskripsikan fitur yang dikombinasi oleh *feature detector* lain. BRIEF merupakan deskriptor yang menggunakan *binary test*. Pada deskripsi atau ekstraksi fitur, ORB mengadopsi metode BRIEF yang didesain ulang agar dapat mendeskripsi fitur pada gambar. Berikut merupakan langkah – langkah yang dilakukan metode ORB untuk mendeskripsi fitur pada gambar:

1. Melakukan binary test pada  $\tau(p: x, y)$ , lalu memberi nilai dengan ketentuan:

*1, jika  $p(x) < p(y)$*

*0, untuk kemungkinan lain*

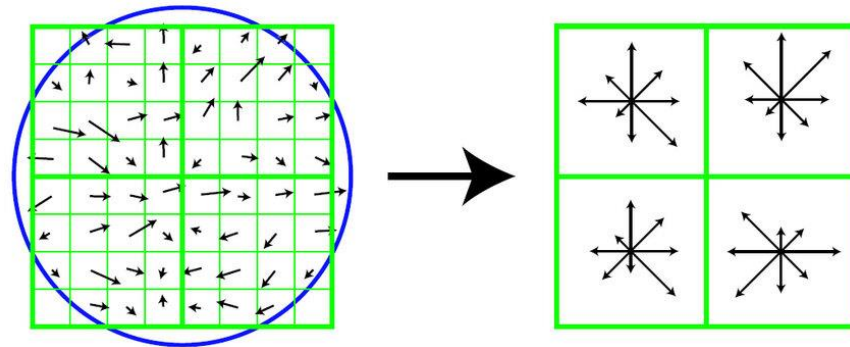
Dimana  $p(x)$  merupakan intensitas piksel yang dilakukan smoothing dari  $p$  pada titik  $x$ .

2. Lalu terapkan rumus BRIEF pada dimensi  $nd$ :

$$f_{nd}(p) = \sum_{l \leq i \leq nd} 2^{i-l} (p: x_i, y_i) \quad (4)$$

3. Untuk setiap  $S * S$ , piksel di smoothing dengan Gaussian dan pilih piksel menggunakan binary test.
4. Melakukan spatial arrangement of the binary test untuk setiap titik piksel. piksel gambar yang di smoothing.

## 2.5. Algoritma SIFT



Sumber [www.researchgate.com](http://www.researchgate.com)

Gambar 2.4 Image Gradient (kiri), Keypoint Descriptor (kanan)

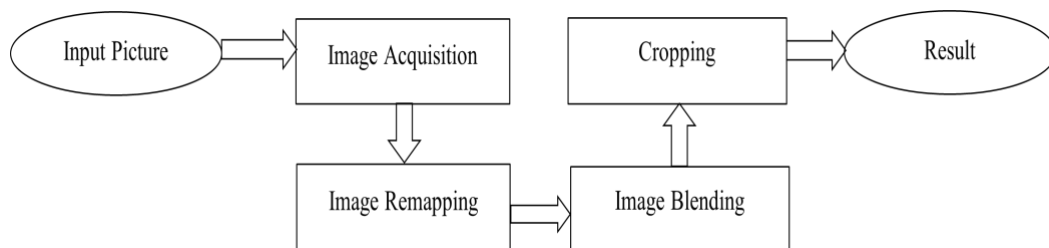
Algoritma SIFT (*Scale Invariant Feature Transforms*) adalah algoritma yang membangun fitur deskripsi dari gambar yang invarian terhadap orientasi dan penskalaan yang seragam, dan tahan terhadap oklusi parsial dan perubahan cahaya (Byrne J, 2017). SIFT sendiri juga menggunakan beberapa algoritma untuk membangun fitur vektor, diantaranya adalah algoritma Gaussian dan algoritma HoG (*Histogram of Gradients*). Mula-mula, SIFT membuat suatu skala gambar dengan menggunakan perbedaan algoritma Gaussian. Algoritma ini membandingkan piksel yang buram di titik gambar yang berbeda (Byrne J, 2017). Fitur ini dievaluasi menggunakan *contrast threshold* dan *discard points below a prespecified threshold*. Diagram Histogram (HoG) dihasilkan untuk tiap fitur yang kemudian digunakan sebagai fitur vektor deskriptor seperti pada gambar 2.4.

Kelemahan SIFT sendiri, yang pertama berkaitan dengan *Gaussian blurring* untuk mencari fitur (contoh, penggunaan rata-rata dengan fungsi kernel untuk memburamkan gambar). Pendekatannya pada skala kasar yang dapat menghapus fitur yang memiliki fitur dengan *noise* dan struktur objek. Kedua adalah penggunaan deskriptor HoG yang memiliki panjang 128 bit, yang membuat kurang efisien secara komputasi.

## 2.6. Image stitching

*Image stitching* diambil dari kata “*Image*” yang berarti “gambar” dan “*Stitch*” yang berarti "Menjahit". *Image stitching*, atau dikenal dengan image mosaik, adalah proses menggabungkan gambar dengan area yang tumpang tindih untuk membentuk gambar dengan tampilan lebar dan resolusi tinggi. (Wang, Z., dan Yang, Z. (2020)). Penjahitan gambar dapat digunakan untuk membuat mosaik panorama yang cantik, yang mana sering dilihat secara interaktif untuk aplikasi seperti pariwisata maya, atau memberikan latar belakang dalam film dan video game (D. Pribadi *et al.*, 2013). Menurut Hafizah Raehana (2022) menyatakan bahwa *image stitching* merupakan proses penggabungan beberapa gambar atau foto dengan bidang area yang saling bertumpukan untuk menghasilkan panorama dengan ukuran sangat besar . Berikut adalah *flowchart image stitching* dalam produksi gambar panorama di gambar 2.5.

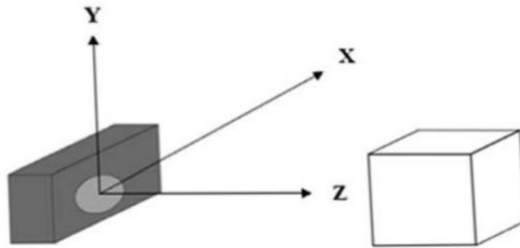
### 2.6.1 Proses Image stitching



Gambar 2.5 Alur Proses Produksi Gambar Panorama

### 2.6.1.1 Image Acquisition

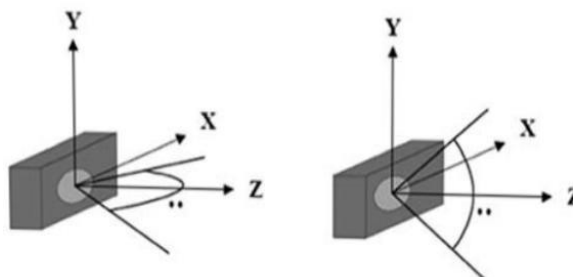
Tahap proses *image stitching* yang harus dilakukan adalah kalibrasi gambar. Hal ini memerlukan pemilihan posisi dan perolehan gambar. Tahap ini memerlukan keputusan (decision) dibuat berdasarkan jenis gambar panorama yang dihasilkan. Sesuai dengan gambar panorama yang diperlukan, metode *Image Acquisition* yang berbeda digunakan untuk memperoleh rangkaian gambar. Berikut adalah sistem koordinat dari kamera pada gambar 2.6



Sumber [www.link.springer.com](http://www.link.springer.com)

Gambar 2.6 Sistem Koordinat Kamera

Gambar 2.6 menjelaskan dimana sumbu Z mengarah ke objek yang diinginkan dan sumbu Y melewati sumbu optik kamera. Sudut pandang kamera dalam arah horizontal dan vertikal menentukan cakupan tiap gambar dalam arah horizontal dan vertikal. Pada gambar 2.7 dimana masing-masing sudut mewakili sudut pandang kamera secara horizontal dan petunjuk arah vertikal.



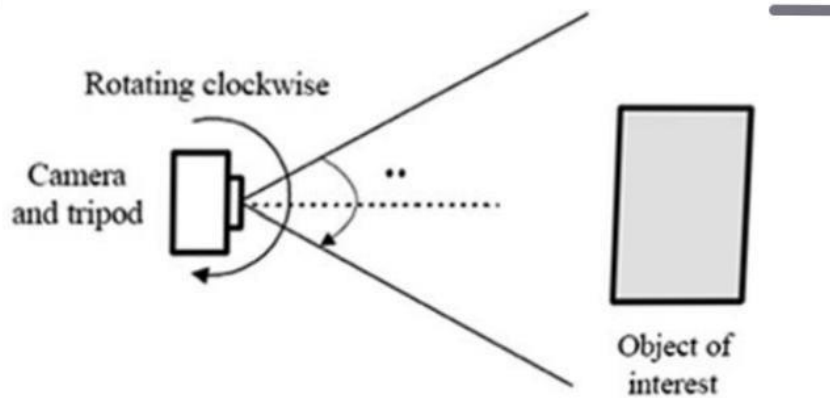
Sumber [www.link.springer.com](http://www.link.springer.com)

Gambar 2.7 Sudut Pandang Horizontal dan Vertikal

#### 2.6.1.1.1 Akuisisi gambar dengan rotasi kamera

Contoh metode mengakuisisi dengan rotasi kamera yakni pertama-tama adalah mengatur kamera dipasang ke tripod dan gambar diambil dengan memutar kamera. Metode akuisisi tersebut, diatur harus rata pada posisi yang dipilih dan tetap pada posisi tetap selama gambar diakuisisi. Setelah mengamankan kamera pada tripod, selanjutnya adalah memfokuskan pada objek yang diinginkan dan diputar dalam satu arah yang dipilih. Satu gambar diambil dengan setiap putaran kamera hingga rentang waktu yang diinginkan. Gambar 2.8 menunjukkan tripod dan kamera untuk akuisisi gambar dengan cara merotasi kamera

Tiap gambar dalam merangkai panorama, sebagian gambar tumpang tindih

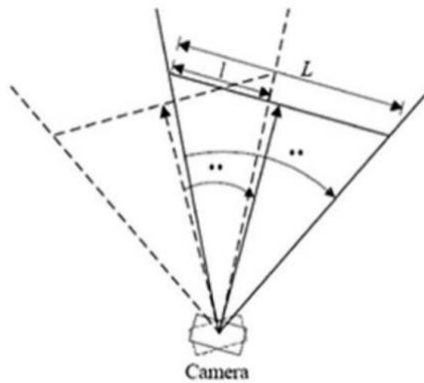


Sumber [www.link.springer.com](http://www.link.springer.com)

Gambar 2.8 Akuisisi dengan Rotasi Kamera

dengan gambar jahitan sebelumnya. Besar area gambar yang tumpang tindih (overlapping) adalah faktor penting dari *image stitching*. Semakin besar area tumpang tindih gambar yang berdekatan digabungkan, lebih mudah dalam langkah penggabungan gambar. Berikut adalah Gambar 2.9 yang menjelaskan ilustrasi dari gambar tumpang tindih





Sumber [www.link.springer.com](http://www.link.springer.com)

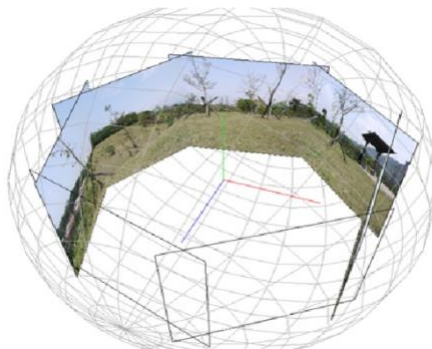
Gambar 2.9 *Overlap* Dua Gambar

Pada gambar tersebut, ‘L’ merepresentasikan lebar antara gambar tumpang tindih dan gambar yang berdekatan. Rasio ‘l’ ke ‘L’ tergantung dari sudut roasi antara gambar yang berurutan dan sudut pandang horizontalnya.

#### 2.6.1.2 *Image Remapping*

Image Remapping merupakan proses mengubah geometri gambar agar sesuai dengan gambar yang berdekatan. Proses *image remapping* ini “membengkokkan” gambar sedemikian rupa sehingga dapat disejajarkan dengan sempurna. Jika tanpa melakukan proses *image remapping*, maka gambar tersebut tidak akan menyatu dengan benar.

##### 2.6.1.2.1 Konversi gambar ke bentuk bola



Sumber [www.link.springer.com](http://www.link.springer.com)

Gambar 2.10 Konversi Gambar ke Bentuk Bola

Gambar 2.10 menjelaskan bahwa gambar dari kamera harus diubah menjadi sudut pandang suatu ruang yang divisualisasikan sebagai proyeksi gambar dari kamera, membalikkan jalur cahaya ke dalam sebuah bola.

#### **2.6.1.2.2 Menyelaraskan Gambar ( *Image alignment* )**

Beberapa teknik dalam menyelaraskan gambar, dapat dideskripsikan sebagai berikut.

##### *1. Feature Based Registration*

*Feature detection* merupakan deteksi fitur yang ciri khas yang menonjol dan terdeteksi. Metode ini dapat digunakan tergantung pada situasi. Metode secara otomatis dapat diterapkan ketika landmark geometri digunakan. Untuk wilayah bisa menggunakan metode segmentasi dimana fitur garis terdeteksi melalui *Canny detector*.

*Feature matching* merupakan fitur pencocokan dimana berkoresponden antar gambar yang ditetapkan. Masalah dapat muncul disini karena terjadinya kesalahan deteksi fitur atau kamera distorsi. *Feature matching* telah dijelaskan di sub-bab sebelumnya menggunakan algoritma ORB.

#### **2.6.1.3 Image Blending**

Setelah tiap piksel dari gambar tersebut sudah melakukan remapping, selanjutnya bagaimana gambar tersebut dipadukan untuk menciptakan panorama/gambar yang atraktif. *Image blending* digunakan untuk mengurangi dan menghilangkan jahitan (proses stitching karena perbedaan pencahayaan), buram, berbayang (karena benda bergerak).

#### **2.6.1.4 Cropping**

Cropping merupakan pemangkasan gambar yang mengacu pada bagian atau area yang tidak diinginkan dari gambar atau panorama. Salah satu dasar dari

*image manipulation* ialah melakukan penghapusan subjek yang tidak diinginkan atau mengubah aspek rasio atau untuk memperbaiki keseluruhan

## 2.7. Web

*Website* adalah halaman yang ditampilkan di internet yang memaparkan informasi tertentu. *World Wide Web*, atau Web, WWW, atau W3, dikembangkan pada tahun 1990 di CERN (Laboratorium Fisika Partikel) di Swiss (Sari et al., 2019). Website adalah struktur hypertextual yang digunakan untuk menampilkan data dalam bentuk teks, gambar, suara, animasi, dan data multimedia lainnya (Efendi, 2017). Website ini umumnya dibangun menggunakan HTML, dan semua format dokumen hyperlink yang dapat diklik—seperti gambar, dokumen, multimedia, dan formulir yang dapat diisi—berdasarkan HTML.

Aplikasi berbasis web adalah aplikasi yang dikembangkan dengan menggunakan bahasa HTML, PHP, CSS, dan JS, yang memerlukan web server dan browser seperti Edge, UC browser, atau Opera untuk dijalankan. Aplikasi web dapat berguna baik jika terhubung internet maupun intranet (Jaringan LAN). Data yang terpusat dan kemudahan akses adalah fitur utama yang membuat aplikasi web lebih populer dan mudah diimplementasikan dalam berbagai bidang kehidupan (Adani 2018).

## 2. 8. SSIM (*Structural Similarity Index Measures*)

Kesamaan struktural antara gambar digunakan sebagai indeks untuk mengukur kualitas gambar. Metode SSIm tidak mempertimbangkan kesalahan. SSIM mengevaluasi kualitas gambar dilihat dari tiga aspek kecerahan, kontras, and kesamaan struktur gambar. Oleh karena itu, fungsi evaluasi SSIM dapat dinyatakan sebagai:

$$S(x, y) = f(l(x, y), c(x, y), s(s, y)) \quad (5)$$

Dimana  $l(x,y), c(x,y), s(x,y)$  mewakili kecerahan, kontras dan kesamaan struktur antar gambar dan  $(x,y)$  masing-masing merepresentasikan referensi gambar dan gambar yang akan dievaluasi.  $S(x,y)$  adalah skor evaluasi untuk SSIM.  $f()$  merepresentasikan hubungan pemetaan fungsi. Tetapkan  $\mu_x, \mu_y$ , kecerahan rata-rata dua gambar,  $\sigma_x \sigma_y$ , nilai standar informasi abu-abu. Kemudian, koefisien korelasi intrakelas timbal balik antara dua gambar didefinisikan sebagai  $\sigma_{xy}$ , sehingga algoritma SSIM didasarkan pada pengukuran kesamaan ketiga komponen ini:

$$\begin{cases} l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{cases} \quad (6)$$

Dimana  $C_1, C_2$ , dan  $C_3$  adalah konstanta untuk mencegah penyebut rendah dalam perhitungan aktual yang menyebabkan ketidakstabilan fungsi. Melalui nilai bobot yang berbeda, tiga komponen di atas merupakan sistem penilaian model evaluasi SSIM:

$$SSIM(x,y) = [l(x,y)]^\alpha [c(x,y)]^\beta [s(x,y)]^\gamma \quad (7)$$

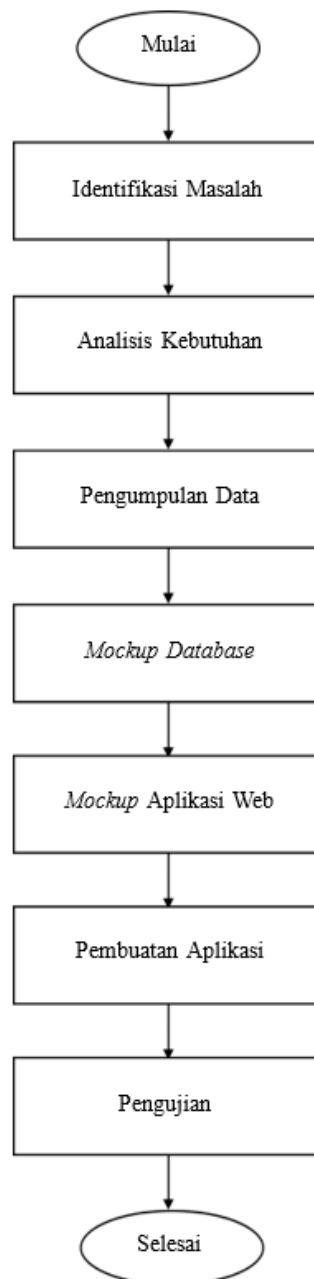
$\alpha, \beta, \gamma$  merepresentasikan bobot nilai dari tiga komponen kecerahan, kontras dan kesamaan struktur. Nilai bobot bergantung dengan persyaratan aplikasi yang berbeda.

## BAB III

### PERANCANGAN APLIKASI

#### 3.1. Alur Penelitian

Proyek ini dibuat dengan tujuan membuat aplikasi berbasis web dengan *framework* Django dalam melakukan penerapan *image stitching* pada data gambar UAV / *drone*. Alur penelitian ini dapat dilihat pada gambar 3.1 sebagai berikut .



Gambar 3. 1 Alur Penelitian

Penelitian dimulai dengan mengidentifikasi masalah dan tujuan yang jelas akan permasalahan yang dibahas. Setelah itu akan dilanjutkan dengan studi pustaka yakni mempelajari dan mendapatkan sudut pandang lain dari penelitian terdahulu yang berkaitan dengan penelitian ini. Selain itu, landasan teori yang berhubungan akan dicari dan dipelajari dari buku, jurnal, *website*, *e-book*, dsb. Hal ini dilakukan untuk mengetahui metode yang tepat untuk menyelesaikan permasalahan yang akan diteliti. Setelah itu, akan dilakukan pembuatan *mockup database* untuk kebutuhan data apa saja yang akan dimasukkan ke dalam aplikasi web. Selanjutnya adalah tahap *mockup* aplikasi web dimana web didesain dengan mengedepankan user friendly. Tahap *mockup* dibuat sesederhana mungkin, dan *user* merasa tidak perlu repot menggunakannya. Pembuatan mockup aplikasi web telah dilaksanakan, maka selanjutnya adalah pembuatan aplikasi web yang dinamakan “*stitch'em*” dengan menggunakan *framework* django dari python dan *database* menggunakan SQLite bawaan dari django itu sendiri. Terkait dengan *database*, aplikasi hanya memerlukan gambar mentah (*raw image*) dan hasil proses *stitching*. Setelah menerapkan *database*, selanjutnya adalah menampilkan *landing page* dari aplikasi web dibuat sesederhana mungkin agar *user* mudah akses aplikasi tersebut. Terakhir yaitu tahap pengujian dimana aplikasi yang sudah dibuat akan diuji dua tahapan, yaitu uji fungsionalitas, dan uji *Ground Truth* dengan mengukur tingkat keselarasan pada gambar menggunakan SSIM. Uji tingkat keselarasan bertujuan untuk membandingkan apakah dari gambar orisinal dengan gambar diproses *image stitching* memiliki kesamaan. Pengujian ini sangat umum terutama ranah *Computer Vision* karena untuk membuktikan bahwa gambar orisinal

Selanjutnya akan dilakukan pengumpulan data yang nantinya akan digunakan untuk proses uji *image stitching* dengan algoritma ORB untuk menggabungkan data gambar UAV dalam 1 kali misi, menjadi suatu peta. Dalam pengambilan data, misi merupakan rencana awal yang akan diambil gambarnya. Pengambilan data gambar, dalam 1 kali misi dengan UAV/*drone* hanya bertahan selama 15 menit. Pengumpulan data sudah dilakukan, maka tahap akhir yakni melakukan pengujian *image stitching* dengan algoritma ORB.

### 3.2. Identifikasi Masalah

Masalah yang ditemukan dalam penelitian sebelumnya , yakni penelitian dari Theo Evan Le Gall (2023) yang membahas tentang aplikasi kontrol drone berbasis web untuk mendesain, mengembangkan dan memvalidasi, yaitu masalah terkait konfigurasi *resolution*, *exposure*, *saturation* pada hasil gambar drone. Jika dari salah satu dari parameter tersebut tidak diatur, maka proses stitching tidak berjalan dan detail dari gambar kadang tidak terlihat. Selain itu, hasil dari stitching muncul dark border karena pengambilan gambar dari drone yang tidak stabil (tergantung posisi *altitude* dan *relative position*).

### 3.3. Analisis Kebutuhan

Pada perancangan dan pembuatan sistem penggabungan data gambar UAV berbasis web dengan algoritma ORB dibutuhkan analisis kebutuhan. Analisis kebutuhan mencakup kebutuhan perangkat keras, dan perangkat lunak baik dari sisi pengguna maupun peneliti. Sehingga nantinya penelitian dapat berjalan dengan lancar.

#### 3.3.1 Kebutuhan Pengguna

Kebutuhan analisis ini diperoleh berdasarkan identifikasi masalah yang telah dijelaskan pada sub-bab sebelumnya, yaitu implementasi *computer vision* menggunakan metode *image stitching* untuk proses penggabungan beberapa gambar yang dihasilkan dari *drone/UAV* menjadi satu gambar yang utuh. Pengembangan *computer vision* bertujuan untuk membantu user yang memiliki data *drone/UAV* dapat dijadikan suatu peta. Sehingga user tidak perlu mengeluarkan biaya lebih dan dapat digunakan oleh user dari kalangan *mid-end* sampai *low-end* komputer. Pengguna cukup mengakses web yang akan peneliti buat untuk menggunakannya.

#### 3.3.2 Kebutuhan Peneliti

Berikut ini adalah beberapa perangkat keras dan juga perangkat lunak yang digunakan oleh peneliti dalam melakukan penelitian ini.

1. Perangkat Keras
  - a. Komputer
  - b. Prosesor: AMD Ryzen 5 5600X 6-Core 3.70 GHz
  - c. RAM: 8 GB
  - d. SSD: 500 GB
  - e. GPU: Radeon™ RX 460
  - f. Sistem Operasi: Windows 11 Pro 64 bit
2. Perangkat Lunak
  - a. Visual Studio Code
  - b. Python 3
  - c. Django
  - d. OpenCV
  - e. Numpy
  - f. Glob
  - g. Imutils

### 3.4. Studi Pustaka

Tahap studi Pustaka akan dilakukan terkait dengan penelitian sebelumnya dan menjadi acuan dalam menyelesaikan masalah dalam penelitian ini. Berbagai macam referensi yang akan digunakan dalam penelitian ini adalah referensi yang berhubungan dengan image stitching menggunakan algoritma ORB. Dalam tahapan ini, bertujuan untuk menambah wawasan dan tahap-tahap proses yang dapat digunakan dalam penelitian. Berikut adalah ringkasan penelitian terdahulu dapat dilihat pada tabel 3.1 sebagai berikut.

Tabel 3. 1 Hasil Ringkasan dari Penelitian Terdahulu

No	Topik	Pengetahuan	Temuan
1.	Research on image feature point matching based on	Algoritma ORB yang dibantu algoritma RANSAC dalam	ORB jika dikombinasikan dengan algoritma



No	Topik	Pengetahuan	Temuan
	ORB and RANSAC algorithm.	ekstraksi feature dan kecocokannya.	RANSAC lebih baik dibanding menggunakan ORB sendiri. (Zhang H., <i>et al</i> 2020)
2.	UAV low-altitude aerial image stitching based on semantic segmentation and ORB algorithm for urban traffic.	UAV ketinggian rendah dengan metode <i>image stitching</i> berdasarkan segmentasi semantik dan algoritma ORB untuk daeran lalu lintas.	Peneliti menemukan bahwa kombinasi informasi semantik dari proses segmentasi semantik dengan algoritma ORB <i>stitching</i> dapat mencapai gambar udara berkualitas tinggi (Zhang G., <i>et al</i> 2022)
3	Design, development, and validation of a web-based control application for drones	Pembuatan aplikasi web yang mampu menggunakan drone dalam mengambil dan merekonstruksi gambar	Pengambilan dan merekonstruksi gambar dengan teknik <i>image stitching</i> (Le Gall., 2023)

Rangkuman dari studi pustaka pada tabel 3.1 konsep pembuatan aplikasi berbasis web ini, komponen yang serupa adalah pada penelitian Le Gall yang mengemukakan bahwa algoritma ORB dapat diimplementasikan ke dalam aplikasi berbasis web dan dapat merekonstruksi gambar dengan teknik *image stitching*. Namun yang menjadi pembedanya ialah dalam, terdapat fitur perencanaan wilayah

terbang *drone* (fotogrametri) dengan sekaligus hasil pengambilan masuk ke dalam *database* aplikasi, lalu diproses dengan teknik *image stitching*

### 3.5. Pengumpulan Data

Pada pengerjaan proyek ini, data yang dikumpulkan dilakukan melalui pengambilan gambar menggunakan *drone*. Pengambilan gambar dilakukan di kota Daerah Istimewa Yogyakarta. Sebelum melakukan pengambilan gambar, terlebih dahulu melakukan suatu perencanaan jalur untuk *drone* mengambil gambar. Selanjutnya, pengaturan ukuran gambar bawaan dari drone sebesar 4000x3000 piksel, Hal ini dapat diubah sesuai kebutuhan masing – masing. Terakhir, perencanaan pengambilan gambar ada dua kali tahap pengambilan.

Perencanaan awal dimulai dari mengambil daerah yang objeknya adalah daerah perumahan dan lapangan. Setelah melakukan perencanaan, *drone* akan tereksekusi dan terbang dengan sendiri sesuai dengan jalur yang sudah ditentukan. Setelah kurang lebih 30 menit, didapatkan 2 folder yang dinamakan Mission 1 dan 2. Dari 2 folder tersebut digabungkan menjadi 1 folder yang isinya 97 gambar berekstensi .JPG dengan total ukuran filenya sebesar 474 MB.

Perencanaan kedua dilakukan di kota yang sama namun objeknya berbeda yaitu di Yogyakarta spesifiknya di jalan tol. Proyek tol terdapat berbagai macam objek teridentifikasi seperti sungai, pepohonan, rumah, dan pepohonan. Pengaturan yang sama dengan pengaturan pada perencanaan awal yaitu dalam pengambilan gambar pada wilayah tdan menghasilkan masing – masing 169 gambar dengan ukuran filenya hampir mencapai 1 GB. Berikut dapat dilihat pada tabel 3.2.

Tabel 3. 2 Daftar Data gambar UAV/Drone

No	Nama Folder	Jumlah Citra	Besar Folder
1	Mission 1 & 2	97 gambar	474 MB

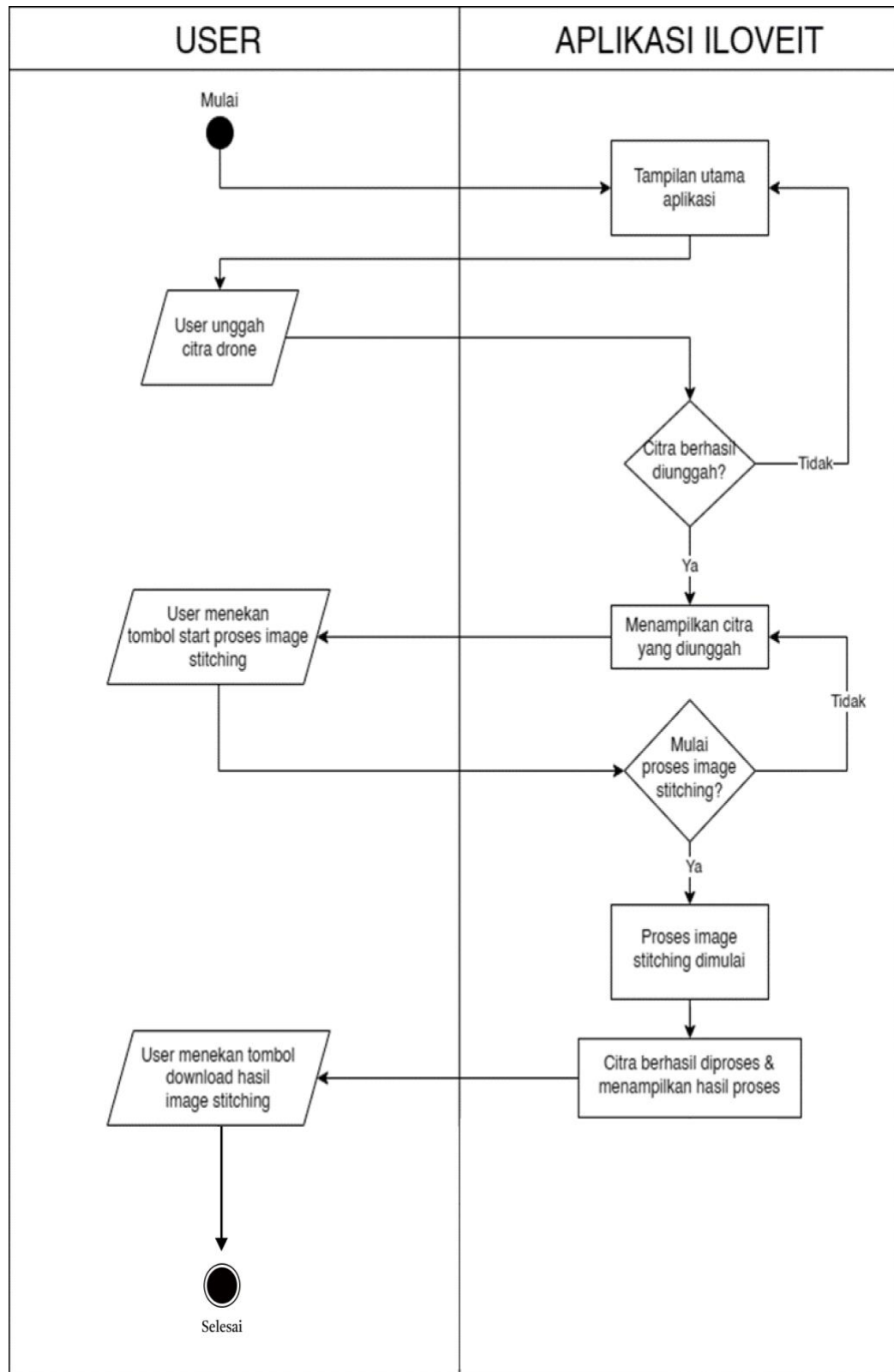
No	Nama Folder	Jumlah Citra	Besar Folder
2	Projek Tol (Mission 1&2)	169 gambar	1.6 GB

### 3.6. Desain Sistem

#### 3.6.1 Alur Aplikasi

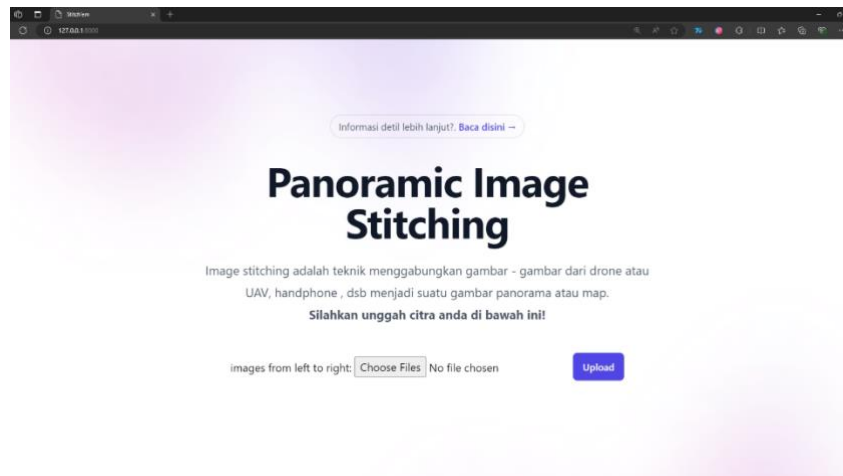
Fungsi utama dalam sistem yang dibangun adalah melakukan penggabungan (*stitching*) gambar UAV/drone menjadi satu gambar peta. Konsep dari aplikasi web ini sendiri adalah sebagai fasilitator. Aplikasi fasilitator memiliki kelebihan yakni dengan desain UI yang sederhana dan responsif. Hal ini sangat mempengaruhi sisi *user* karena mudah dipahami. Saat *user* menggunakan aplikasi “iloveit” ini, *user* yang memiliki gambar UAV/drone atau memiliki potongan – potongan gambar yang ingin digabungkan, maka halaman utama (*landing page*) ditampilkan. Lalu, *user* mengunggah gambar sesuai dengan jumlahnya. Sebelum mengunggah gambar, *user* wajib memastikan gambar tersebut berekstensi .jpg, .png, .jpeg karena aplikasi web menerima *file* berekstensi .jpg, .png, dan .jpeg. Selain itu maka aplikasi web tidak bisa memproses *stitching*. Jika *file* berhasil terunggah, tampilan tetap ada di *landing page* dan memberikan informasi hasil *file* yang diunggah akan muncul, dan *user* melakukan pengecekan jika memang diperlukan. Pengecekan bertujuan jika ada *file* yang berbeda atau tidak sesuai dengan ekstensi. Setelah melakukan pengecekan, selanjutnya *user* menekan tombol start untuk melakukan proses *image stitching*. Tampilan halaman proses image stitching terdiri dari informasi jumlah *file* yang diupload, dan waktu pemrosesan. Waktu proses tergantung dari berapa banyak *file* yang diunggah. Setelah proses selesai, maka tampilan halaman selanjutnya adalah menunjukkan hasil *stitching*. Hasil *stitching* tersebut, *user* dapat mengunduh melalui web tersebut. Berikut gambar 3.2 adalah activity diagram yang menjelaskan alur aplikasi web dengan *user*

### 3.6.2 Mockup Aplikasi dan Pembuatan Aplikasi



Gambar 3. 2 Activity Diagram User

Setelah mengetahui alur aplikasi beserta penjelasannya, selanjutnya adalah gambaran dari aplikasi sesuai dengan alur yang sudah dijelaskan sebagai berikut.



Gambar 3. 3 Landing Page

Pada gambar 3.3 merupakan tampilan halaman awal aplikasi image stitching. Halaman awal terdapat beberapa tombol yakni “*about*”, “*contact*”, dan “*Select your image*”. Dari kolom *header*, tombol “*about*” adalah informasi terkait aplikasi yang dibuat dan tujuan dari pembuatan aplikasi. Tombol “*contact*” adalah untuk menampilkan informasi kontak yang bisa dihubungi. Tujuannya adalah jika ada *error* maupun *bug*, dapat dilaporkan diinformasi kontak tersebut. Selanjutnya adalah tombol “*select your image*” bertujuan untuk user memilih gambar yang sudah disiapkan, lalu diunggah ke aplikasi tersebut.



Gambar 3. 4 Halaman file sudah terupload

Selanjutnya, jika user sudah mengunggah. Pada gambar 3.4 merupakan tampilan halaman setelah user mengunggah gambar tersebut. Proses *stitching* bisa dilakukan jika user menekan tombol “start”. Proses yang akan dilalui ada 3 tahap yaitu *pre-processing*, *feature matching* dan *image blending*. Tahapan tersebut membutuhkan waktu tergantung berapa banyak gambar yang diunggah.



Gambar 3. 5 Hasil Stitching

Selanjutnya, pada gambar 3.5 hasil dari proses stitching. Dapat disimpulkan bahwa alur aplikasi menjelaskan terkait alur bagaimana interaksi user menggunakan aplikasi tersebut. Lalu mockup aplikasi menjelaskan terkait tampilan halaman dan fungsi-fungsi tombol dari aplikasi web tersebut

### 3.7. Rancangan Pengujian Sistem

Pengujian aplikasi yang dibangun akan menggunakan dua buah penilaian yang berupa uji fungsional *Image Stitching* dan uji perbandingan dengan metode lain menurut 10 partisipan yang diuji coba menggunakan perangkat lunak mereka masing masing.

### 3.7.1. Rancangan pengujian Fungsionalitas Image stitching

Pengujian fungsional sangat diperlukan bagi developer agar mengetahui bahwa penerapan *image stitching* pada *website* bekerja. Pengujian dilakukan mulai dari *user* mengunggah beberapa gambar dari drone, lalu saat memproses dengan *image stitching* dengan ORB, sampai hasil gabungan gambar menjadi map.

Tujuan dari pengujian ini yaitu memastikan bahwa *user* dapat mengunggah beberapa gambar sesuai dengan format dan ukuran yang sesuai, lalu verifikasi algoritma ORB dapat mendeteksi dan dapat mencocokkan fitur dari gambar *drone*, menilai kecepatan dan kecocokan dari proses *stitching*, terakhir yaitu memastikan hasil gabungan gambar dapat membentuk suatu peta.

### 3.7.2. Rancangan pengujian Ground Truth

Pengujian *Ground Truth* adalah pembelajaran dalam *machine learning* dimana data yang dianggap benar. Seperti contohnya ada satu gambar yang menggambarkan objek binatang misalnya kucing dan gambar lain yaitu kuda. Dengan visual manusia dapat memahami bahwa dari kedua gambar tersebut adalah gambar yang berbeda.

Berbeda dengan cara kerja dari komputer itu sendiri, karena komputer tidak dapat membedakan dari dua objek binatang tersebut. Komputer perlu mengerti bagaimana membedakan dua objek binatang tersebut. Maka dari itu pengujian *ground truth* diperlukan.

Rancangan pengujian *ground truth* dalam *image stitching* diperlukan untuk membuktikan bahwa kinerja komputer dalam memproses *stitching* dapat serupa dengan gambar asli (seperti contohnya adalah map). Proses *ground truth* dilakukan dengan mencari data asli seperti map atau gambar panorama, setelah mendapatkan datanya maka dilakukan pemotongan gambar satu persatu bertujuan untuk distitching di aplikasi yang sudah dibuat. Setelah gambar terpotong, maka proses *stitching* dapat dilakukan dan hasilnya akan dibandingkan dengan data asli.

Dalam melaksanakan *ground truth* ada dua pengujian yang dilakukan yaitu uji SSIM (*Structural Similarity Index Measure*). Uji SSIM dilakukan dengan tujuan untuk mengukur kemiripan dari kedua gambar. Mengukur kemiripan kedua gambar dimulai mengukur pencahayaan gambar, kontras, dan struktur dan membandingkan nilai – nilai pada dua gambar tersebut Hasil pengukuran tersebut, uji SSIM terdapat dua parameter yang dapat dipenuhi yaitu gambar berbeda (diberi nilai -1) atau gambar identik (diberi nilai 1). Sebaliknya, Hal yang perlu dilakukan untuk melakukan Berikut adalah contoh gambar 3.7.2 membandingkan kedua gambar dengan menggunakan SSIM

MSE: 16256.49, SSIM: 0.66



[https://github.com/imamun93/Image-Similarities-using-SSIM/blob/master/blog5\\_ssim](https://github.com/imamun93/Image-Similarities-using-SSIM/blob/master/blog5_ssim).

Gambar 3.6 Contoh perbandingan kedua gambar

Contoh kedua gambar memiliki jenis yang sama namun yang membedakan adalah tiap keturunan hewan memiliki karakteristik yang berbeda. Seperti halnya bentuk muka, warna bulu, bentuk telinga, dsb. Seperti pada gambar 3.7.2 mendapatkan nilai SSIM sebesar 0.66, yang mana artinya kedua gambar tersebut dianggap sama.



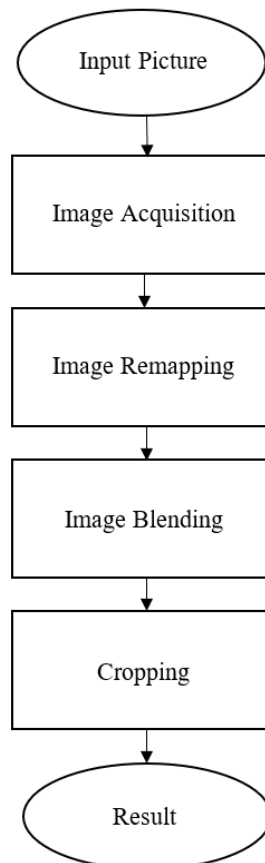
## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Sistem Image Stitching

*Image stitching* digunakan untuk membuat suatu gambar panorama atau gambar beresolusi tinggi. Secara tidak sadar, sistem ini sering digunakan melalui berbagai macam cara, seperti kamera dari *handphone*, kamera *drone*, dsb. *Image stitching* dibagi menjadi 3 cara, yang pertama pengambilan gambar, lalu diproses dan digabungkan.

Proses dalam image stitching menggunakan bahasa Python dan library OpenCV. Berikut adalah konsep dasar dari *image stitching* dan bagaimana kinerjanya pada gambar 4.1.



Gambar 4. 1 Sistem Image Stitching

Alur sistem dari *image stitching* dimulai dari mengumpulkan gambar (*image acquisition*). Sebelum mengumpulkan gambar, teknik akuisisi gambar harus ditentukan apakah mengambil berdasarkan perputaran pada kamera atau tanpa perputaran kamera (pergerakan pada kamera). Setelah menentukan pengambilan gambar, selanjutnya mengekstrak fitur dan menyatukan gambar dengan teknik *remapping* agar menghasilkan *keypoint*. Teknik *remapping* menggunakan algoritma ORB dimana awal mulanya mengekstrak suatu fitur gambar lalu dicocokkan dengan FLANN (*Fast Library Approximate Nearest Neighbor*). FLANN berperan dalam mencari hasil titik fitur paling mirip yang dihasilkan ORB. Setelah fitur gambar sudah dicocokkan dan digabungkan, maka tahap filter pada gambar dilakukan (*image blending*). Hasil gambar yang sudah dicocokkan terdapat perpotongan halus maupun kasar, tergantung dari gambar yang diunggah. Contoh dalam pengambilan gambar, terdapat perbedaan cahaya, selain itu jika menggunakan *drone*, faktor angin dapat berpengaruh karena dalam pengambilan gambar tidak konsisten, dsb. Maka dari itu, *image blending* berperan dengan mengatur dari gambar tersebut, seperti menghaluskan perpotongan gambar yang kasar, mengatur pencahayaan agar selaras, dsb. Terakhir setelah gambar sudah selaras, maka pemotongan (*cropping*) dilakukan. *Cropping* yaitu memotong daerah yang tidak diperlukan agar menjadi suatu gambar kesatuan. *Cropping* telah dilakukan maka didapatkan hasil *image stitching* seperti gambar panorama atau gambar peta/map.

*Image stitching* dapat digunakan bermacam - macam cara, contohnya adalah bagaimana cara membuat suatu map dari *drone*, bagaimana cara memproses gambar menjadi panorama, dsb. Sistem dari ORB itu sendiri diantaranya.

Proses *image stitching* akan dilakukan menggunakan class `cv::ORB`. Class `cv::ORB` ini dibangun menggunakan libraries *Computer Vision* yang memperbolehkan kita mengekstraksi fitur ORB untuk melakukan *image stitching*. Algoritma ORB terutama modul pada openCV `cv::ORB::create()` memiliki beberapa parameter untuk mengontrol proses *image stitching* yang diantaranya sebagai berikut.

```
1. cv.ORB.create([,nfeatures[, scaleFactor[, nlevels[,  
edgeThreshold [,firstLevel [,WTA_K  
[,scoreType[,patchSize[,fastThreshold]]]]]]]])
```

Dimana,

1. *nfeatures* menentukan batasan jumlah fitur yang akan dipertahankan.
2. *scaleFactor* menentukan tingkat penurunan piramida, dan memengaruhi skor pencocokan fitur.
3. *nlevels* menentukan jumlah tingkat piramida.
4. *edgeThreshold* menentukan ukuran tepi dimana fitur tidak terdeteksi.
5. *firstLevel* adalah menetapkan tingkat piramida untuk gambar sumber.
6. *WTA\_K* menentukan jumlah titik di tiap elemen deskriptor BRIEF berorientasi.
7. *scoreType* menentukan metode untuk merangking fitur.
8. *patchSize* menentukan ukuran *patch* yang akan digunakan oleh BRIEF.
9. *fastThreshold* menetapkan ambang batas untuk algoritma FAST.

Proses *image stitching* akan dibantu oleh algoritma yang lainnya. Jia Yinjiang dan rekan lainnya memperkenalkan suatu filter penajaman, yakni BF (*BruteForce*) dan RANSAC (*Random Sample Consensus*) sehingga algoritma dapat mengekstrak lebih banyak *feature points* disaat memproses *stitching*. Algoritma ORB, adalah algoritma yang memiliki fitur gambar *real-time* yang representatif. Berdasarkan fitur, ORB lebih unggul daripada algoritma populer lainnya dalam hal kecepatan sekaligus memastikan *feature points* memiliki invarian rotasi dan skala.

## 4.2. Perancangan Aplikasi

Penggabungan gambar *drone* dengan metode *image stitching* dapat membangun sebuah sistem berbasis web yang baik dan perlu juga mengingat tujuan awal dari Tugas Akhir yaitu membuat web yang dapat menerima gambar *drone* dan menggabungkannya dengan metode *image stitching* menggunakan ORB. Sebelum masuk pada pembahasan *image stitching* berbasis web, perlu diketahui bahwa

sebelum melakukan stitching, ukuran gambar dari *drone* harus diperkecil karena batas dalam melakukan *stitching* tidak lebih dari 100 Mb. Menurut Ng. W. H. (2021) tujuan dari gambar diperkecil adalah saat *pre-processing*, gambar dengan ukuran 1920x1080 piksel menjadi 640x360 piksel dengan tujuan mempercepat kalkulasi dalam *image stitching*. Setelah diperkecil, gambar akan diubah menjadi *grayscale* karena dapat memperoleh proses komputasi yang lebih cepat. Maka dari itu, untuk melakukan *pre-processing*, dataset dari UAV 4000x3000 piksel diperkecil menjadi 600x800 piksel seperti gambar berikut.

Dimensions	4000 x 3000	Dimensions	600 x 800
Width	4000 pixels	Width	600 pixels
Height	3000 pixels	Height	800 pixels

Gambar 4. 2 Dataset UAV original (sisi kiri) diperkecil (sisi kanan).

#### 4.2.1 Model Forms

Sebelum merancang suatu web, diperlukan *upload form* dalam mengunggah foto sampai hasil proses *stitching*. *Form* menggunakan *method* “POST” dan *enctype* “*multipart/form-data*”. Kedua, perlu mengkoneksikan suatu *file data*. *File data* ditangani secara terpisah dengan *file data* bentuk normal dalam *framework* Django. Dalam aplikasi ini, menggunakan *FileField* karena *FileField* memiliki beberapa atribut yang berguna seperti contoh “*upload\_to*” untuk memberikan nama direktori untuk menyimpan gambar yang disimpan dan “*validators*” untuk memastikan gambar yang diunggah sesuai dengan *extension*, *size*, dsb.

Model *form* dari aplikasi yang dibuat dapat dilihat pada gambar 4.2... yang terlampir di bawah ini dengan menggunakan bahasa Python.

```
1. class ImageForm(forms.Form):
2.     image_field =
forms.FileField(widget=forms.FileInput(attrs=None))
3.
```

Setelah model dibuat, selanjutnya adalah membuat suatu *function* ke dalam *file* *views.py* untuk menyimpan jika gambar sudah distitching. Umumnya file ‘*views*’ dapat berkomunikasi langsung dengan file ‘*forms*’ tanpa harus membuat suatu *database*. *Form* yang dimaksud adalah *Regular Forms*, dimana *form* dibuat

tanpa perlu model dan hanya perlu menyebutkan *fields* dan *validation* dari *form* untuk mengelola data dari *input user*.

#### 4.2.2 Image Pre-processing

```
1. def stitch_images(images, f_method="orb",  
matcher="flann", lowe_ratio=0.75, max_threshold=100)
```

Fungsi diatas mengikuti suatu argumen '*images*' yaitu menerima *list of image* yang akan distitch, lalu *f\_method="orb"* adalah *method feature detection* dengan opsi pilihan bisa '*sift*' maupun '*orb*', selanjutnya *matcher="flann"* adalah *method* untuk menyamakan *keypoint* dari *feature detection* yang dihasilkan dengan opsi pilihan '*brute\_force*' dan '*flann*', *lowe\_ratio* adalah nilai untuk memfilter kecocokan *keypoint*.

##### 4.2.2.1 Fetch image

###### 4.2.2.1.1 Konversi Grayscale

Tahap awal sebelum proses pengolahan gambar adalah mengkonversikan gambar dari RGB ke *grayscale*. Tahap ini diawali dengan pembacaan gambar masukan. Sistem akan berjalan apabila terdeteksi dua buah gambar masukan pada direktori. Kedua gambar tersebut adalah gambar uji dan gambar acuan. Masing-masing gambar masukan merupakan gambar RGB, ketika kedua gambar terdeteksi, maka dilakukan proses konversi RGB ke *grayscale*.

```
1. for image in images:  
2. grey = cv.cvtColor(img, cv.COLOR_BGR2GRAY)  
3. if method is "orb":  
4.     kp, des = orb.detectAndCompute(grey, None)  
5. features.append((kp, des))  
6. Return features
```

Keadaan awal gambar terdiri dari 3 lapisan matrik, yaitu *R-layer*, *G-layer* dan *B-layer*. Kondisi gambar dengan 3 lapisan matrik RGB ini membuat proses-proses dalam pengolahan selanjutnya harus memperhatikan ketiga lapisan matriks

tersebut. Bila setiap proses perhitungan harus memperhatikan 3 lapisan, maka harus dilakukan 3 perhitungan yang sama hal ini memerlukan waktu yang relatif lama, sehingga untuk meminimalisir proses perhitungan pengolahan gambar yakni dengan penggunaan gambar dalam kondisi *grayscale*. Berikut adalah gambar yang *digayscale*.



Gambar 4.3 Gambar yang Digayscale

## 4.2.3 Feature Extraction

### 4.2.3.1 Deteksi Keypoint

Hasil konversi dari RGB ke *grayscale* diikuti dengan *deteksi keypoint*. Fitur seperti gelembung (*blob-like feature*) digunakan untuk mendeteksi *keypoint*. Prosesnya melibatkan pembentukan piramida gambar menggunakan *box filter* sebagai pendekatan dari turunan parsial kedua *Gaussian*. Selama pembentukan *scale space*, gambar asli dikonvolusi dengan *box filter* yang diubah ukurannya sesuai dengan gambar, membentuk *scale space*. Langkah berikutnya adalah mencari ekstrema dari determinan matriks *Hessian* dengan membandingkannya dengan tetangga-tetangganya. Calon fitur kemudian dilokalisasi pada setiap ruang skala (*scale space*) dengan menggunakan *non-maximum suppression* terhadap ekstrema dari determinan matriks *Hessian*. Hasil akhir dari tahap ini adalah titik-

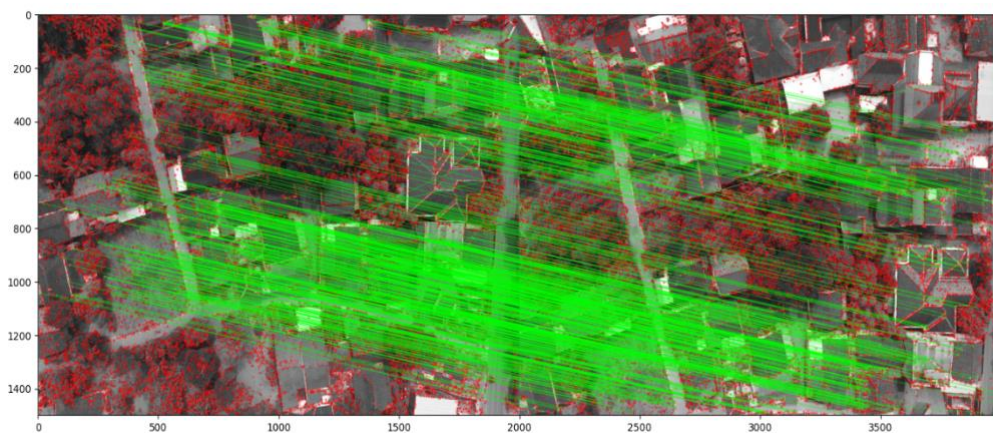
titik acuan atau *keypoint* yang menunjukkan adanya fitur gelembung (*blob-like feature*)



Gambar 4.4 Deteksi *keypoint*

#### 4.2.3.2 Dekripsi Keypoint

Pendeskripsian *keypoint* diawali dengan menghitung respon *Haar-wavelet* terhadap sumbu-x dan sumbu-y dengan titik-titik di lingkungan tetangganya pada radius  $6s$  di sekitar titik-titik fitur, dengan  $s$  adalah skala pada titik fitur yang terdeteksi.



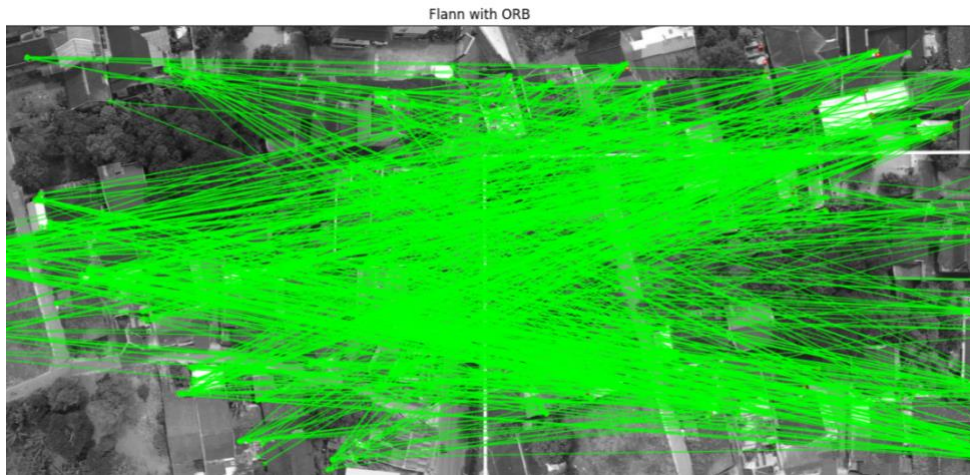
Gambar 4.5 Dekripsi *keypoint*



#### 4.2.3.3 Algoritma FLANN

*Keypoint* yang terdeteksi akan dieliminasi sebagian dengan metode reduksi data. Setelah jumlah *keypoint* pada gambar sudah tereduksi, dilakukan proses deskripsi *keypoint* dengan deskriptor ORB. *Keypoint* pada kedua gambar masukan yang sudah dideskripsi kemudian dicocokkan menggunakan FLANN.

```
1. flann = cv.FlannBasedMatcher(index_params,  
search_params)  
2. all_matches = flann.knnMatch(des_a, des_b, k = 2)  
3. good_matches = Lowe_ratio_test(all_matches,  
Lowe_ratio)  
4.
```



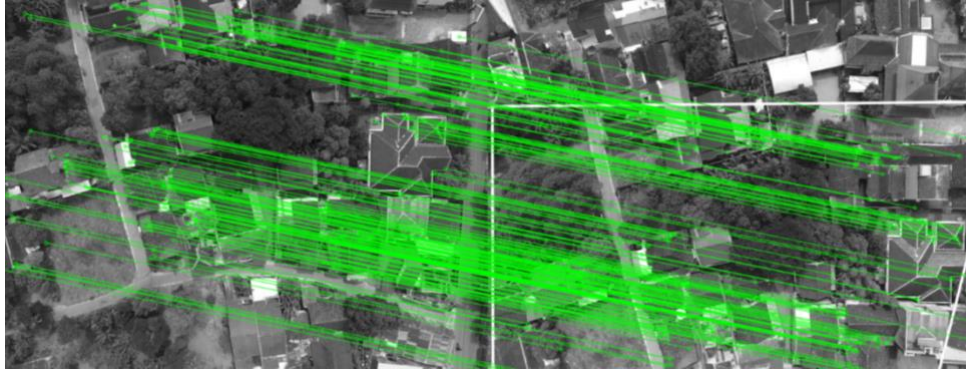
Gambar 4.6 Hasil Pencarian kecocokan Keypoint Gambar 1 dan 2

Metode *Fast Library for Approximate Nearest Neighbors* (FLANN) adalah pustaka yang digunakan untuk pencarian cepat dan perkiraan tetangga dalam ruang berdimensi tinggi. Pustaka ini berisi berbagai algoritma yang efektif untuk menemukan nilai tetangga terdekat, dengan hasil parameter yang optimal bergantung pada kumpulan data yang digunakan. Fitur ORB mencakup *keypoint*, *deskriptor*, dan vektor. Sebuah gambar dalam basis data dapat memiliki banyak klaster untuk fitur ORB. Berikut adalah gambar hasil pencarian kecocokan *keypoint* dari FLANN.



#### 4.2.4 Homography Estimation (RANSAC)

Hasil penyelarasan gambar digunakan sebagai dasar untuk menemukan matriks homografi menggunakan pendekatan RANSAC. Berikut adalah gambar hasil dari mencocokkan keypoint setelah proses mencari kecocokan dengan FLANN.



Gambar 4.7 Hasil Mencocokkan Keypoint dengan RANSAC

```
1. (homography, status) = cv.findHomography(points_a,  
points_b, cv.RANSAC, max_threshold)
```

Untuk menganalisis *keypoint* dalam proses pencocokan *keypoint* dapat menggunakan pendekatan RANSAC. RANSAC dapat mengidentifikasi *inlier* dan *outlier*. *Inlier* merupakan garis yang terbentuk dari titik, sementara *outlier* merupakan titik dan tidak membentuk garis. Titik-titik *inlier* dari poin acak kemudian dipilih oleh RANSAC untuk digabungkan menjadi sebuah garis. RANSAC sangat bergantung pada hasil dari tahap sebelumnya untuk memastikan bahwa pasangan keypoint yang dicocokkan benar-benar akurat

#### 4.3. Hasil Pengujian

Proses pengujian pada Tugas Akhir ini akan dilakukan dengan menggunakan laptop dengan spesifikasi CPU 2.50GHz processor, 12GB RAM, dan Windows 11 Pro sebagai OS. Pertama, tahap yang dilakukan adalah menguji fungsional dari aplikasi *website image stitching* itu sendiri, dimulai dari awal unggah gambar sampai dengan hasil menggunakan algoritma ORB. Kedua, mengevaluasi *features extraction* (penghasil *keypoint*) dan *keypoint* yang sama menggunakan RANSAC. Proses yang dilakukan, menggunakan 2 tipe data yang resolusinya diperkecil yaitu 1000x750 resolusi dan 600x800 resolusi. Hasil dari uji

yang dilakukan mengikuti rancangan pengujian yang dapat dilihat pada sub-bab yang akan dibahas.

Sistem yang dirancang dapat menggabungkan dua gambar inputan dari *drone* agar diperoleh hasil penggabungan gambar dengan cakupan yang lebih bagus. Gambar udara masing-masing diproses menggunakan algoritma ORB.

```
1. def stitch_images (images, f_method="orb",  
matcher="flann", Lowe_ratio=0.75, max_threshold=100):  
2.
```

#### 4.3.1 Hasil Pengujian Fungsionalitas Image Stitching

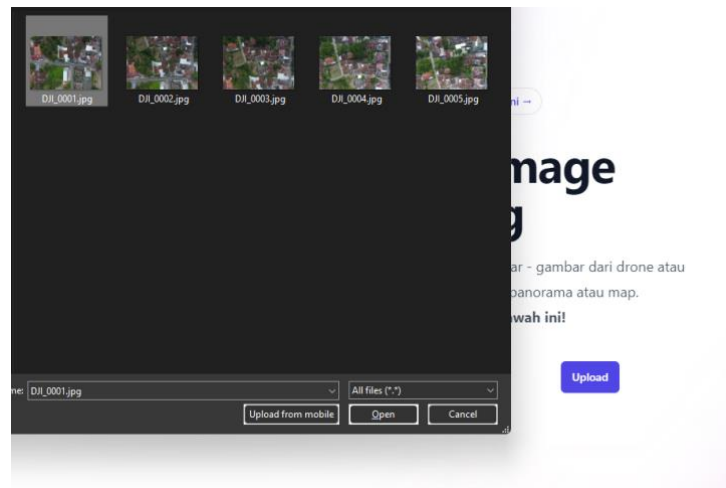
Pengujian fungsional dimulai dari seluruh fitur dari program yang sudah dibangun dan dicoba sekaligus ditunjukkan dengan tujuan dapat dipahami alur dari *image stitching* pada aplikasi web yang dibangun ini. Uji fungsional akan dimulai dari bagaimana proses pengguna melakukan *image stitching*.

Jalannya proses pengujian yaitu melakukan pengunggahan foto sampai hasil menjadi map. Berikut adalah salah satu contoh gambaran dari proses aplikasi web *image stitching* yang terlampir berikut ini.



Gambar 4.8 Landing page dari aplikasi

Pada gambar 4.8 di atas adalah halaman awal dari aplikasi web *stitching* dimana *user* melakukan *input* gambar yang sudah diperkecil setidaknya sebesar 75% agar bisa *distitching*. Seperti yang sudah dijelaskan pada sub-bab sebelumnya bahwa ada keterbatasan dalam aplikasi ini yaitu konsep *stitching* dapat diproses jika gambar diperkecil sekaligus besar file tidak lebih dari 100 Mb. Berikut adalah gambar 4.9 yang menjelaskan sisi *user* mengunggah gambar yang sudah diperkecil.



Gambar 4.9 User upload gambar yang sudah diperkecil

Selanjutnya adalah setelah sudah mengunggah gambar, maka proses *stitching* dapat dilakukan. Berikut gambar 4.10a sampai 4.10c adalah hasil *stitching* dari aplikasi web



Gambar 4.10a Hasil Unggahan pertama



Gambar 4.10b Hasil Unggahan Kedua



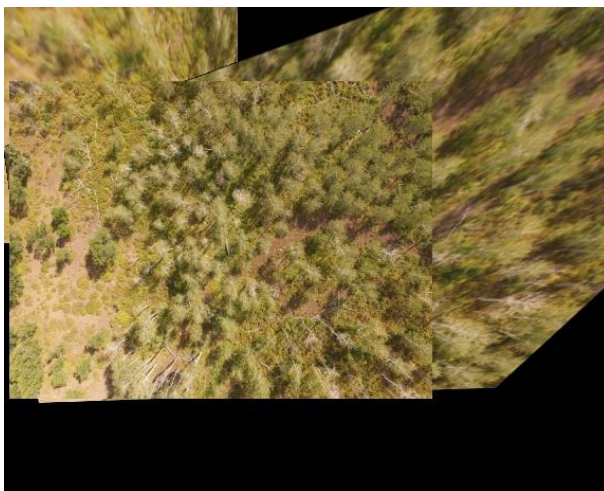
Gambar 4.10c Hasil Unggahan Ketiga

Gambar di atas adalah hasil *stitching* dari gambar drone. Namun ada beberapa hal yang dijelaskan seperti warna hitam yang mengelilingi gambar dari drone. Hal ini terjadi karena disaat melakukan *warping*, piksel bagian tepi gambar diambil terinterpolasi dengan piksel *background* warna hitam.

Selain hasil *stitching* di kawasan perkampungan, berikut adalah hasil dari *dataset* proyek tol sesuai dengan ketentuan sub-bab sebelumnya.



Gambar 4.11 Hasil *Stitching* Proyek Tol Yogyakarta



Gambar 4.12 Hasil *Stitching* Data Gambar Hutan

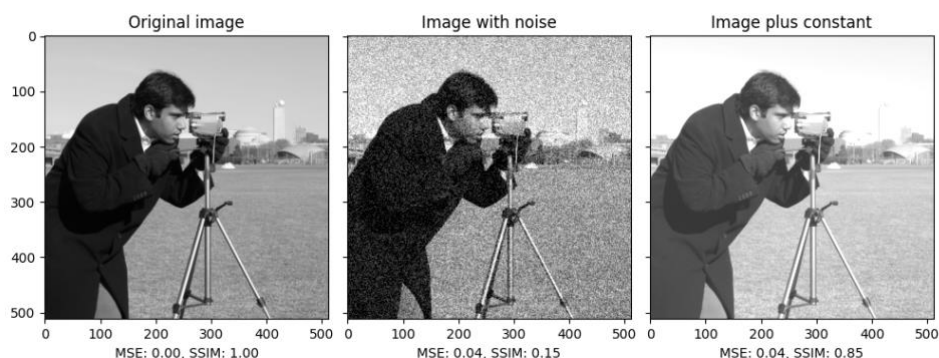


Gambar 4.13 Hasil *Stitching* Data Gambar Pabrik

Dataset dari file proyek tol gambar 4.11, dataset hutan pada gambar 4.12 dan dataset pabrik pada gambar 4.13 didapatkan hasil menyerupai map, meskipun terdapat gambar yang terdistorsi dan tidak selaras. Hal ini menyebabkan *overlapping* antar gambar tidak memadai, dan algoritma ORB kesulitan untuk mengkalkulasi titik yang cocok. Selain dari algoritma, dalam pengambilan foto dari sudut pandang yang berbeda, terutama jika pemandangan berisi objek pada jarak yang bervariasi, hal ini menyebabkan masalah dalam pencocokan.

### 4.3.3 Hasil Pengujian *Ground Truth*

Tahap akhir yaitu uji *Ground Truth* dengan menggunakan metode SSIM (*Structural Similarity Index Measure*). Pengujian dilakukan dengan membandingkan gambar panorama/map yang asli dengan hasil gambar panorama/map dari aplikasi web 'iloveit'. Tujuan dari pengujian ini untuk mengetahui hasil gambar panorama/map dari algoritma ORB secara struktural apakah memiliki kesamaan dengan gambar panorama/map yang asli. Perbandingan struktural dilihat mulai dari kualitas objek gambar seperti contohnya gambar memiliki *noise*, *contrast*, *blurry image*, dan perbedaan resolusi gambar. Berikut adalah gambar 4.14 yang menjelaskan dasar uji *ground truth* SSIM dari yang sudah dijelaskan.



Sumber: [https://scikit-image.org/docs/stable/\\_images/sphx\\_glr\\_plot\\_ssim\\_001.png](https://scikit-image.org/docs/stable/_images/sphx_glr_plot_ssim_001.png)

Gambar 4.14 Dasar Pengujian *Ground Truth*



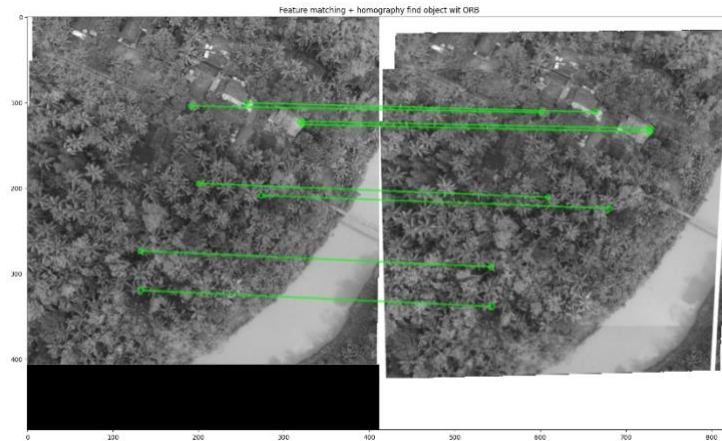
Langkah – langkah yang dilakukan dalam uji *ground truth* adalah mencari gambar panorama/map utuh, lalu melakukan duplikat gambar tersebut dan potong satu persatu dengan ukuran resolusi 1x1, usai melakukan pemotongan gambar, selanjutnya yaitu proses *stitching* ke aplikasi berbasis web yang sudah dibuat, dan hasil gambar *stitching* akan dibandingkan dengan gambar panorama/map asli. Kedua gambar tersebut akan dibandingkan secara struktural maupun menghitung persentase tingkat kecocokan dari kedua gambar. Berikut adalah gambar 4.15 menggambarkan data berupa map permukiman yang akan dibandingkan dengan hasil proses *stitching* dari aplikasi web “*stitch'em*”.



Gambar 4.15 Bagian atas adalah map asli dan Bagian bawah adalah map hasil proses stitching

Jika gambar *stitching* yang dihasilkan oleh aplikasi web terdapat *black border*, maka hal ini berpengaruh terhadap uji SSIM. Pada gambar 4.15 hasil dijelaskan bahwa SSIM berpengaruh terhadap cahaya, kontras, dan struktur dari gambar. Maka dari itu, penting untuk menghilangkan *black border* terlebih dahulu agar ekspetasi kalkulasi SSIM dapat relevan. Selain menghilangkan *black border* pada gambar, uji SSIM tidak bekerj jika dimensi dari kedua gambar tidak sama dan dimensi kedua gambar tersebut harus sama. Sebaliknya jika gambar hasil dari

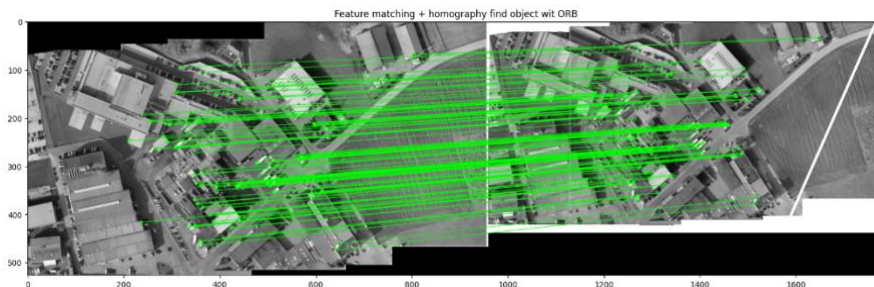
*stitching* tidak terdapat *black border* dan dimensi kedua gambar sesuai, maka hasil kalkulasi SSIM lebih relevan. Berikut gambar 4.16, gambar 4.17, dan gambar 4.18 adalah data yang diambil dari Yogyakarta yang diberi nama ‘projek tol’, ‘mission 1’, dan data ‘Quarry’.



Gambar 4.16 Pencocokan Gambar Asli dan Gambar Hasil *Stitching* Projek Tol

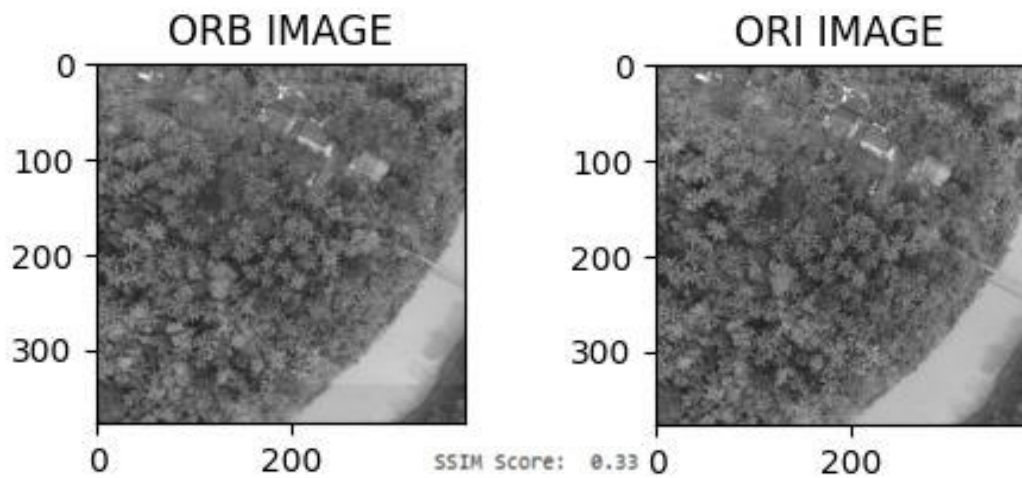


Gambar 4.17 Pencocokan Gambar Asli dengan Gambar Hasil *Stitching* Mission 1

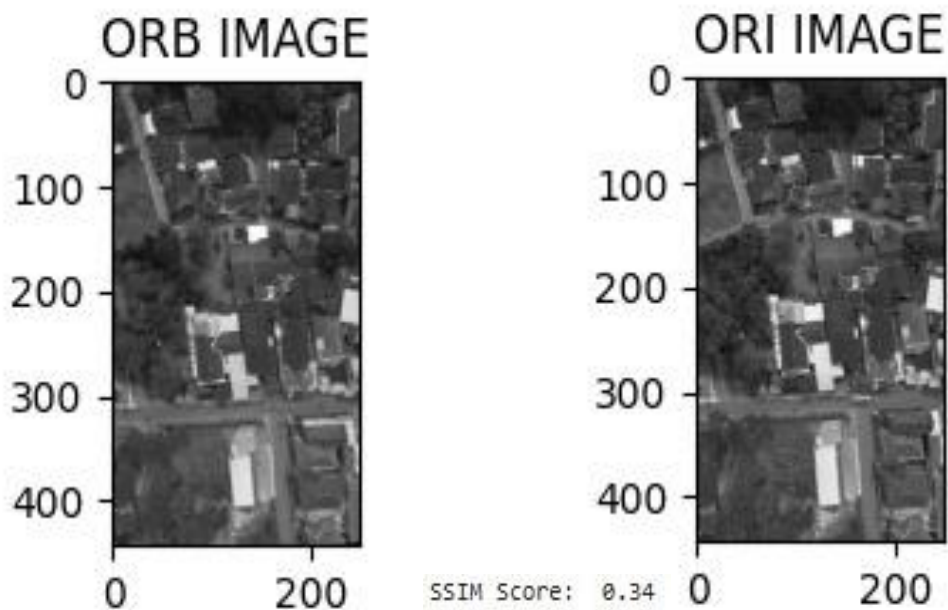


Gambar 4.18 Pencocokan Gambar Asli dengan Gambar Hasil *Stitching* Quarry

Pada gambar di atas (gambar 4.16, 4.17, dan 4.18) menunjukkan bahwa pencocokan gambar asli dengan gambar hasil *stitching* terdapat beberapa kesamaan *feature*. Namun hasil homografi RANSAC, kedua gambar tersebut menunjukkan bahwa dalam mencocokkan *feature* yang sama, tiap objek pada gambar berbeda – beda. jumlah kecocokan *feature*. Selain itu, terdapat perpotongan pada gambar *stitching* dikarenakan perbedaan pencahayaan saat pengambilan. Berikut gambar 4.19, gambar 4.20, dan gambar 4.21 adalah hasil uji SSIM dengan data dari hasil pengumpulan data dan tambahan data dari eksternal website *kaggle.com*.

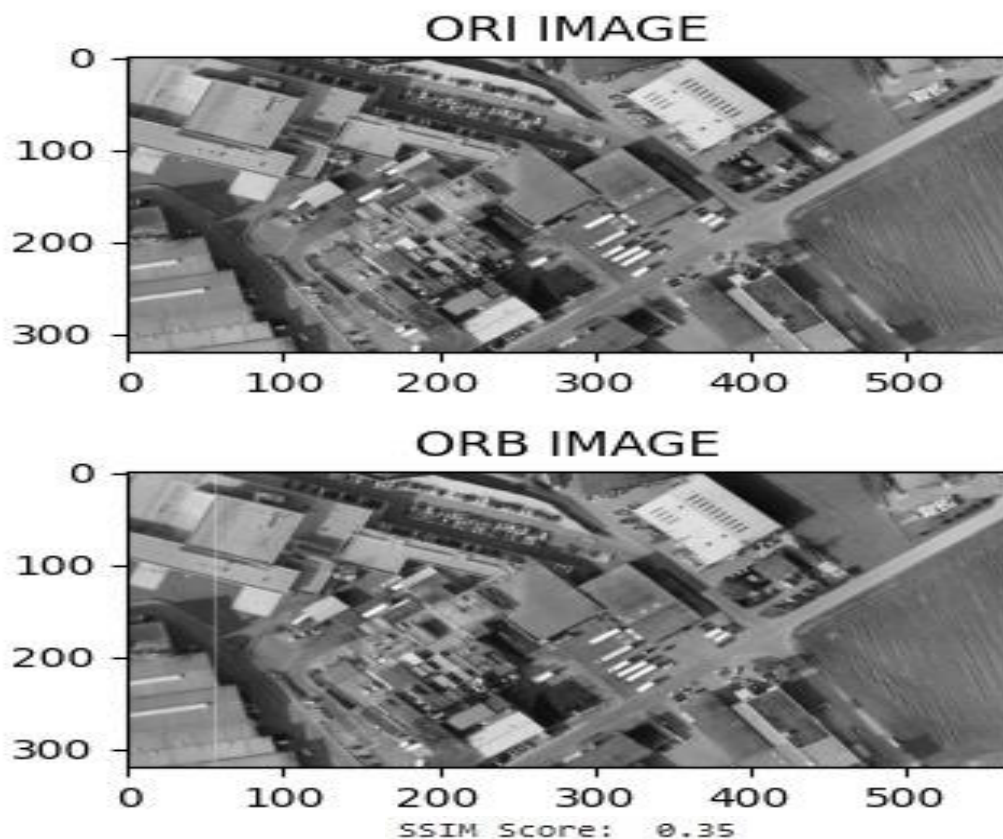


Gambar 4.19 Hasil SSIM projektol Yogyakarta sebesar 33%



Gambar 4.20 Hasil SSIM mission1 Yogyakarta sebesar 34%





Gambar 4.21 1 Hasil Uji SSIM Data Quarry Dari Kaggle Sebesar 35%

Pada gambar 4.19 sampai dengan 4.21 membuktikan bahwa cahaya, kontras, dan struktur sangat berpengaruh. Dilihat pada gambar 4.19 menghasilkan sebesar 33% dan gambar 4.20 sebesar 34% dan gambar 4.21 sebesar 35%. Dapat disimpulkan bahwa hasil uji SSIM dilihat dari pencahayaan, kontras dan struktur kurang baik karena jauh dari nilai parameter yang sudah ditetapkan SSIM. Berikut adalah tabel 4.1 yaitu hasil analisis dari ketiga poin gambar di atas sebagai berikut.

Tabel 4. 1 Hasil Analisis Tiga Objek Gambar Berbeda

<i>Nama Data (dimensi)</i>	<i>Feature Point</i>		<i>Matched</i>	<i>Waktu</i>	<i>Persentase</i>
	<i>Gambar Asli</i>	<i>Gambar Stitching</i>	<i>Feature Point</i>	<i>yang Dibutuhkan</i>	<i>Uji SSIM</i>
Quarry (825 x439)	500	500	127	0.75 detik	35%
Projek tol (412x407)	500	500	11	1 detik	33%
Mission 1 (476x544)	500	500	33	1.37 detik	34%

Tabel 4.1 memaparkan hasil dari analisis dimulai dari ekstraksi jumlah *feature point* kedua gambar, jumlah *matched feature point* dari hasil ekstraksi, lalu waktu yang dibutuhkan dalam mencocokkan *feature points* kedua gambar dan persentase SSIM dilihat mulai dari pencahayaan, kontras maupun struktur gambar. Pada kolom ‘Nama Data’ dimana terdapat deskripsi dimensi. Uji SSIM sangat sensitif terhadap dimensi pada gambar. Oleh karena itu, sebelum melakukan uji SSIM, terlebih dahulu mencocokkan dimensi gambar asli dengan gambar hasil *stitching*. Selain mencocokkan dimensi, gambar harus dipotong (*cropping*) karena pada hasil *stitching* dari aplikasi memiliki *dark border*. *Dark border* sangat berpengaruh karena mempengaruhi nilai uji SSIM sekaligus nilainya tidak relevan.

Selanjutnya *feature* menghasilkan masing – masing sebesar 500 *keypoints* karena dari *library* OpenCV sendiri, terutama *nFeatures* secara umum dipasang batasnya sebesar 500. Lalu hasil ekstrak *feature points* tersebut, dicocokkan dengan menggunakan FLANN (*Fast Library for Approximate Nearest Neighbor*) yang menghasilkan *keypoint* yang sama dan jumlahnya banyak, lalu mengeliminasi *outlier keypoints* menggunakan konsep homografi dengan algoritma RANSAC (*Random Sample Consensus*) dari tiap gambar karena mempengaruhi kecepatan proses komputasi maupun proses *stitching*. Pada tabel 4.1, terutama pada kolom ‘waktu proses’ didapatkan bahwa proses yang paling cepat yaitu pada data ‘Quarry’ yang diambil dari website *kaggle* dengan waktu proses sebesar 0,75 detik.

Dari tabel 4.1 dapat disimpulkan bahwa algoritma ORB diterapkan ke aplikasi berbasis web mampu melakukan *stitching* gambar cukup baik, waktu komputasi cepat. Namun, pada tahap uji *ground truth* menggunakan konsep SSIM mendapatkan nilai yang kurang baik sebesar 33% sampai dengan 35% dengan batasan orientasi proses *stitching* berlaku horizontal atau vertikal.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Perancangan sistem penggabungan gambar dengan metode *image stitching* dapat disimpulkan bahwa algoritma ORB dapat memproses dan diimplementasikan ke dalam aplikasi web yang *real-time* dan dapat memvisualisasikan hasil peta geografis. Namun ada beberapa aspek yang perlu diperhatikan bahwa kualitas dari gambar harus diperkecil karena untuk mempercepat komputasi saat *pre-processing*

#### **5.2 Saran**

Saran untuk pengembangan sistem penggabungan gambar dengan metode *image stitching* menggunakan ORB adalah menyempurnakan sistem *image stitching* itu sendiri karena untuk saat ini aplikasi berjalan secara *local* dan belum bisa *publish* secara umum, lalu pengembangan dari kode *image stitching*, logika kode yang ditulis hanya dapat menggabungkan secara horizontal atau vertikal. Maka dari itu perlu pengoptimalan dan evaluasi kode dari aplikasi *image stitching* berbasis web ini.

## DAFTAR PUSTAKA

- Adani, R. Muhammad. 2018. Jenis Aplikasi Berbasis Web Beserta Contoh Penerapannya. Jakarta : Kencana
- Byrne, J., Laefer, D. F., & O'Keeffe, E. (2017). Maximizing feature detection in aerial unmanned aerial vehicle datasets. *Journal of Applied Remote Sensing*, 11(2), 025015-025015.
- Chang, Y. (2023). *Drone-based panorama stitching: A study of SIFT, FLANN, and RANSAC techniques* (Master's thesis, Universitat Politècnica de Catalunya).
- Cholik, C. A. (2021). Perkembangan teknologi informasi komunikasi/ICT dalam berbagai bidang. *Jurnal Fakultas Teknik Kuningan*, 2(2), 39-46.
- Dewanti, F., & Sumiharto, R. (2015). Purwarupa Sistem Penggabungan Foto Udara Pada UAV Menggunakan Algoritma Surf (Speeded-Up Robust Features). *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 5(2), 165-176.
- Goh, J. N., Phang, S. K., & Chew, W. J. (2021, December). Real-time and automatic map stitching through aerial images from UAV. In *Journal of Physics: Conference Series* (Vol. 2120, No. 1, p. 012025). IOP Publishing.
- Le Gall, T. E. (2023). *Design, development, and validation of a web-based control application. for drones* (Master's thesis, Universitat Politècnica de Catalunya).
- Mango, C. A. (2019). *Pembuatan sistem informasi geografis peta bidang tanah berbasis web mapping* (Doctoral dissertation, ITN MALANG).
- Ng, W. H. (2021). *Image Stitching Of Aerial Footage* (Doctoral dissertation, UTAR).

- Oktaviano, R., Ripanti, E. F., & Pratiwi, H. S. Implementasi Image Stitching pada Aplikasi Virtual Tour Bandar Udara Internasional Supadio. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, 9(3), 381-392.
- Risyda, F., Nuryamin, Y., Id, A., & Id, Y. Y. C. (2023). Perancangan Sistem Informasi Manajemen Invoice Menggunakan Generator Freamwork Django-Python Berbasis Website Pada Pt. Lampuind Tekno Elektrik. *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, 10(1), 207-220.
- Sari, E. P., Wahyuni, A., & Narti, N. (2019). Sistem Informasi Sekolah Berbasis Web. *Indonesian Journal on Software Engineering (IJSE)*, 5(1), 87–94.
- Surahman, A., Wahyudi, A. D., Putra, A. D., Sintaro, S., & Pangestu, I. (2021). Perbandingan Kualitas 3D Objek Tugu Budaya Saibatin Berdasarkan Posisi Gambar Fotogrametri Jarak Dekat. *InfoTekJar: Jurnal Nasional Informatika Dan Teknologi Jaringan*, 2, 296–301.
- Widiyaningtyas, T., Prasetya, D. D., & Wibawa, A. P. (2018, October). Web-based campus virtual tour application using orb image stitching. In *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* (pp. 46-49). IEEE.
- Zhang, G., Qin, D., Yang, J., Yan, M., Tang, H., Bie, H., & Ma, L. (2022). UAV low-altitude aerial image stitching based on semantic segmentation and ORB algorithm for urban traffic. *Remote Sensing*, 14(23), 6013.
- Zhang, H., Zheng, G., & Fu, H. (2020, November). Research on image feature point matching based on ORB and RANSAC algorithm. In *Journal of Physics: Conference Series* (Vol. 1651, No. 1, p. 012187). IOP Publishing.
- Zhou, X., Zhang, H., & Wang, Y. (2017, July). A multi-image stitching method and quality evaluation. In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)* (pp. 46-50). IEEE.